Zoe的Python星际之旅

Zoe



目錄

Zoe的Python星际之旅

星际之旅，即将启航。
我们一起，穿越时空隧道，探索好玩的Python吧。

在这里记录，Zoe从零开始的Python旅途。

成长、探索、乐趣。

记录：

- 一个 gitbook 小小私人教程
- 一个 配合课程每周开发任务的目录树

0MOOC

~ 此目录收集, 课程进入时的体验

进展

- 151010 Zoe更新原文六篇
    - Week0:旅途开始
    - 缘起：为何开始
    - 入学三问:从这里启航
    - 初心：是什么打动了我
    - 开始使用MarkDown
    - 提问的智慧
- 150924 大妈创建

Week0:旅途开始

不是学编程，而是学习思维

- 学着用一种新的视角去看待世界，新的思维方式去面对世界。

游戏精神，让生命有趣

- 看到好玩的邮件，我觉得我真的喜欢这样的旅途。
- 编程，好玩。其实就是这么简单的开始。

写作，让自己更看清自己

- 勿忘初心。 写着写着，好像就越明白自己想什么了
- Markdown真好用，让思路更清晰。

提问也要认真

- 问之前要自己思考，尝试解决。
- 经过思考后，更能从他人的回答中获得启迪。
- 提问的时候，要有清晰的表达。
- 大妈总是可以反弹出很多问题回来，来回几次后，这种风格也会让我问之前会多想想。习惯其实就是这样培养的。

邮件列表的讨论方式

- 邮件的方式更适合深入的讨论。
- 思想更有体系，而不是碎片化的，而且好的答案可以方便的常常回味，汲取养分。

缘起 - 为何开始

> 你是编程小白，渴望学会编程，退缩、动摇，但坚持三个月后能做啥？——人生苦短，Python当歌

靠近一点点

2015年，还剩下不到100天。想起年初给自己许下的新年誓愿，两个很简单的想法，但是好像今年忙东忙西，离我当时许下的愿望还是那样的遥远。

今年就快要过去了，明年的我会许下同样的愿望吗？
十年后的我呢，是不是还是有同样的愿望，但是却一直没有付诸行动？

我可以再靠近一点点吗？

五年前我没有开始做这件事，那，我能从今天开始吗？
今天，看上去是我余生里最好的时机。

迷茫

我很迷茫。
在我不喜欢的领域工作，我觉得每天都很痛苦。
生活真的是这样吗？我觉得不是，但是我一下子又不知道要怎么蹦出去。
夜晚突然醒来，看着窗外，感觉生命的流逝，会很清晰的感觉到，我不想这样继续下去。

如果，想要改变，那我要去做一些以前不会做的事情，很少做的事情。
我想，我未知的道路，未知的世界，就藏在这些我没做过的事情里，往这个方向走，我会找到新的线索。

我尝试过很多次，也放弃过很多次。
我每次都会欣喜的发现，世界总是给了我很多新的窗口，新的机会，但也会发现自己到后来莫名其妙的又一次次放弃，回到原地。
如果不是因为喜欢，我不会忍不住一次次想要去尝试。
但是，这背后还有一股力量，我并不是很清楚具体是什么，但是却让我又停止。
一股推力和一股阻力在互相纠结，我想再想办法多使一些力气，度过这一关。

另一种尝试

有人和我说过，对待我现在的这份工作，要么就特别勤奋上进，往上爬；要么就自己轻松一点，过简单的小日子。
而我却纠结，我不想上进，也不想轻松。
后来明白，不想往上爬，因为他们的生活其实并不是我喜欢的，只是别人口中听上去很光鲜，但是却无法真正触动我，我会觉得我只是在随波逐流。
我也不想每天不想事混日子，我觉得很浪费，我觉得内心里有一团火，还没有好好燃烧，不想就被磨灭。
我才发现，我想要第三种生活，去投入的探索我更喜欢的事情。

我不想过完这一生，才发现自己只是在扮演别人眼中的乖乖女。
为了在别人眼中的形象，忙着做别人布置的任务，好累。

我已经长大了，我其实已经有能力，也有资源去做一些自己想做的事情了，不是吗？

看到这期报名启动，我又心动了。
年底比较忙，我不在北上广，除此之外，好像也不是那么的难。
没有完美的机会，就让我再任性一下，多探索一下，送我自己一份礼物吧。

渴望

当我拿到船票，开始旅途，我觉得我像又回到了一个欢呼雀跃的小孩子。
踏上探索未知的旅途，我很开心，我觉得自己又活过来啦。

我明白，接下来我还会碰到很多困难，很多纠结，但是我能感觉我的眼睛又重新亮了起来。

之前的我还担心自己会不会学不下去了，每周能抽出这么多时间来吗？
但当真正开始的时候，我发现其实内心里有一种渴望，有时甚至因为时间太少还有点恋恋不舍，这真的有点出乎我的意料。

给我自己一点空间，让自己自由的成长吧。
没有环境，就为自己创造出一个小小环境来，好好呼吸，好好成长。

星际旅途，即将启航。
我渴望这自由的旅途。

入学三问 - 从这里启航

我想知道什么？

> 我现在的基础如何，为什么想学编程？我来参与开智学院的编程课程，希望学到什么，希望课程结束之后，自己可以做成怎样的事情？

为什么想学编程？

- 想学编程，是因为这是我从小其实就想要去做的事情，但是总是时断时续，没有坚持下来。
- 想到年初给自己定下的新年誓愿，不想这一年过去后，觉得还是没有改变，所以想在最后的三个月里加油一把，让这一年结束的时候不会遗憾。
- 我的人生里花了那么多时间去做别人要求的事情，我也想花点时间去学一些自己喜欢的东西，让自己开心一下。

希望学到什么

- 更大的视野。总觉得现在的视野比较狭窄，基本上就是两点一线的生活，有点浑浑噩噩的过日子。但是我觉得，其实在这里互联网+英语的时代，世界其实给了我很多可能，我想去打开这扇窗，看到更大的世界，做点有趣的更酷的事情。
- 解决问题的思维方式。面对问题常常迷茫，纠结，然后就卡住了。希望自己能够有更结构化的思维方式，更有条理的去分析和解决自己遇到的问题，而不是在茫茫的信息碎片中迷失了，忘记了，然后某一天重新在来过，一直在这里打转转。能够把自己的问题，自己拥有的素材、信息以一种更清晰的方式呈现出来，能够更好地利用现有的资源去通往自己想去的地方。
- 快乐和投入。混日子感觉很空虚，我觉得安逸很无聊，我也不想在现有的岗位往上爬，我觉得我想去我更想去的地方。我想重新找回认真做一件事情很投入很充实也很快乐的感觉。
- 转行。想离开这个行业，去做点自己更喜欢的事情，人生苦短，希望自己早点成长为有足够实力离开这个行业的人，有足够实力去做自己喜欢的事情的人。

希望课程结束后能做成怎样的事情

- 对问题的思考和加工更深入，对信息的组织更有条理。能够更有逻辑的解决生活中碰到的问题。自己搜集的信息，自己写的日记也更体系化。
- 建立自己的一个小小网站，作为自己的创作空间，音乐，写作，编程，都是很有意思的事情，不是么。

- 尝试做一个小的app，能够在自己泄气的时候，鼓励自己要倾听内心的声音。
- 尝试把一些自动化的工作，都自动化完成。
- 认识一些好玩的人，学着和大家一起做一些有趣的事情。

我期待学到什么？

> 对于我希望学到的东西，如果自学，可以如何学到？希望开智课程如何助益自己学到这些东西？

对于我希望学到的东西，如果自学，可以如何学到？

- Zed Shaw的Learn Python the hard way和那本好玩的自动化的教程。
- Udacity
- 自己写日记，搜索，思考，钻研。
- Google

希望开智课程如何助益自己学到这些东西？

- 受到更多人的不同视野的启发，感受到身边优秀的人的影响，更有学习动力。
- 百思不得其解的时候，有人能够请教，或者指引我一个方向。
- 更系统化的学习资源。有些东西可能是我这个层面的看不到的，或者完全没有想到的。能够有一张探索地图，让我慢慢去发掘。
- 交流和支持。让我觉得我不是一个人在战斗，和大家共同成长。

如何证明我学到了什么？

> 怎样的结果能说明我已经掌握了自己想学的内容？

- 给自己搭建了一个创作平台，把自己创作的东西可以放进去。——blog或github有关的空间
- 写东西更有条理和逻辑。笔记的记录有经过二次整理，而不是复制和粘贴。——从笔记到整理成手册
- 面对问题可以拆分，并且逐个攻破，而不只是记录感想。——以后写日记不是发泄，而是尝试更多不同的思路，而且更有逻辑和条理。
- 每天的重复任务自己可以写一些小程序自动化。——自动化的小程序
- 生活更有冲劲。拿出更多的时间去做些自己喜欢的事情，而不是当别人眼中的乖乖女。也不是每天消遣。——能够感受到内心更充实，眼神更明亮。

初心 - 是什么打动了我

1. 成为自己想成为的人

我觉得很打动我的一点，是说到成为自己想成为的人。
可能之前说到学习，说到验收成果，很容易想成是完成一个布置的任务，通过一个什么样的测验，却没有想过自己想要做什么，想成为一个怎样的人。
不是一直都说要自由的吗？但是当这自由交到手上的时候，又有些迷茫，有些忐忑，有些兴奋。
我很喜欢这种给人带来无限可能的感觉，好像卸下心中的负担，重新感觉到我又自由了，这种感觉很有趣。
学着感受自由探索的感觉，一切其实就在并不遥远的地方，伸手就可以触动。 我喜欢的生活，其实并没有那么遥远。

> 参考链接：什么是“课”

2. 用一个新的维度看世界

原来，很多觉得不方便不顺手的事情，懂一点编程，就可以自己去尝试改变；
原来，看上去好像很cool的编程，真正做起来也没有那么难；
原来，这件事情还有我以前没有想到的思考维度，不是没有可能，而是之前的我没有看到机会;
原来，这件事情并没有那么困难，只是我并没有去系统的理清它的脉络...
我一直觉得在这个时代，互联网+英语，给了我们太多的机会和可能，我能感受到它们的强大力量，但是却不知道如何去使用它们。
我现在做的事情，只是一种新的尝试，尝试着从一个新的维度去看待这个世界，以及这个世界赋予我们的机会和无限可能。
我觉得这条路上会有很多不一样的发现。

> 参考链接：大妈答疑：如何用编程让大脑二次发育？

3.享受编程的快乐

说了太多正儿八经的话，我觉得我都快忘记那种最初的感觉了啦，就是享受编程的快乐。
还记得最开始接触编程的时候，真的觉得这就是个超酷的玩具，而且所需要的一切东西，居然用一台小小电脑就可以满足。它可以创作出很多很炫很酷的东西，做很多好玩的小玩意。
是呀，最初的感觉就是，编程很好玩。

好玩！
玩性大发，一时兴起，试着去创作一些自己觉得好玩有趣的东西来，做出来自己都觉得超酷超棒，这种感觉很好玩。
我觉得最该记得的，就是这样一种快乐的感觉。
尝试的快乐，探索的快乐，创作的快乐，游戏的快乐。

> 参考链接：大妈对编程学习的建议

开始使用Markdown

为什么要使用Markdown

- 让写作和排版的过程分离，专注于写作
- .md是纯txt文件，跨平台使用
- 可以让自己写作更有逻辑，更有条理
- 文章输出好看且方便

> 学习材料：
> 献给写作者的 Markdown 新手指南

Markdown 私人教程

背景

安装

配置

使用

体验

提问的智慧

为什么要认真提问

- 可以得到更好的帮助。
- 自己在提问前有思考，看到大家的回答会更有启发。
- 好的问题所引起的讨论，阅读都是一种享受。

如何认真提问

- 描述清楚自己想解决的问题，想达到的目标。
- 写清楚自己是在什么情境下遇到的问题。
- 问之前自己尝试解决，并写上自己做了哪些尝试，结果如何。
- 写问题思路清晰，排版舒适。

> 参考资料：
> 《提问的智慧》
> 提问的智慧—提问前

Week1：搭建自己的学习环境

敢于交流，敢于提问

- 尝试第一次在邮件列表的提问，特别受启发。觉得这帮小伙伴们太棒了。有这么棒的小伙伴在身边，不认真交流真的好浪费！
- 大妈没有读心术，问问问！其实问出去了，他的回答都很给人启发的。
- 思维碰撞更有趣。自己的视角其实是有一定的局限的，所以有些东西自己琢磨容易钻牛角尖。感受一下和他人交流脑洞大开的感觉，其实很high的。

Git和Github

- 对小白来说，是有门槛的。不过是真的好用，值得好好去学习的。
- 对它有所了解后，思路就从原来的，为什么要用git这么复杂的东西，变成了为什么不用git这么好用的东西，哈哈。

如何学习

- 要操作，不能光看视频。不然打瞌睡，还容易忘。
- 输出是更好的输入。自己写过一遍后，尤其是用自己的方式重新组织语言后，确实印象会更深一些。
- 磨刀不误砍柴工。花点时间看些入门的教程，形成更体系的思维，确实比临时不停的google更好用。

Gitbook

- 第一次有写书的感觉很兴奋。
- 原来写一本自己的小小的书，也没有那么难的。
- 写教程和笔记相比，教程会更有体系一些。

我觉得我选择参加这个课程是对的

- 眼睛更有神了，也不会觉得无聊混日子了，觉得每天有东西要思考，要学习，而且是自己感兴趣的，觉得很有动力。
- 现在觉得上班其实不是累，而是看不到自己的成长和进步，总觉得原地打转，很疲惫。

因为上了一天班很累的情况下，学下编程居然又让我恢复的感觉，出乎我的意料呀。

- 编程给了我一个新的解决方式，我觉得无聊的是个人都能做的事情，其实可以让它们自动化，让自己从重复无聊的工作中解放出来，做点更有意思的事情。
- 总觉得每天在学习一些新的东西，写一些新的东西，探索一些新的世界，创造一些新的东西。很充实。而且这也是我更喜欢的方向。
- 大妈很棒，小伙伴们很棒，有太多值得学习的地方。

Sublime

Sublime

Download Sublime

Launch your editor from the command line

1. Find where the subl command is located. subl comes with Sublime, so it should be in the directory where Sublime is located for you.
   For many people, this is /Applications/Sublime\ Text\ 2.app/ Contents/SharedSupport/bin.
   To test this, run ls /Applications/Sublime\ Text\ 2.app/ Contents/SharedSupport/bin.
   You should see the subl command listed.
   If you get the error No such file or directory, Sublime is located somewhere else for you and you'll need to find it.

2. If you do not have a file named .bash_profile in your home directory, create it.
   Because the name of this file begins with a period, it will not appear in most file navigators or when you run ls.
   Instead, run ls -a to see if you have the file.

3. Add the line export PATH=$PATH:/Applications/ Sublime\ Text\ 2.app/Contents/SharedSupport/bin to the end of your .bash_profile.
   If subl was in a different directory for you in step 1, use that directory.

4. Close and re-open your terminal.

5. Typing subl in the terminal should now open Sublime.

参考资料

Setting up Sublime@Udacity

Command Line学习笔记

> 更新日志
>
> - 20151016 增加iomooc卡片 命令行工具下载

还记得"社交网络"那部电影吗?你记得facebook的工程师是怎么与电脑进行交互的吗?

他们用的是一个叫作命令行的东西.
通过命令行,你可以用最快的速度把你的指令告诉电脑. 虽然过去二十年里,图形界面已经有显著发展,但命令行依然是程序员首选的与电脑进行互动的方式,你还没有用过吗?现在就试一试吧.

下载

马上下载最好用的命令行工具:

假如你用的是 Windows 系统---
建议安装运行 Cygwin

假如你用的是 Mac OSX 系统---
建议安装运行 iTerm2

安装好之后建议添加到 Dock 那里,这样可以更方便地找得到. 也可以通过 Spotlight 搜索 iTerm2 来调出.

假如你用的是 GNU/Linux 系统---
用系统自带的命令行工具即可 :)

快捷键

- ctrl c 中断
- ctrl d 退出 ends of file , not any more input coming
- ctrl r 搜索命令
- Tab Completion - 自动完成到可以完成的最远的地方 Tab Again

命令

- ls

- curl http://udacity.github.io/ud595-shell/stuff.zip -o things.zip
- brew install cowsay
- date
- host udacity.com
- echo you rock
- expr 2+2
- uname -a
- history
- hostname - Domain Name System (DNS)
- uptime
- unzip
- cat - concatenate (reading short files)
- wc - word count (lines, words, bytes)
- diff- 比较文件的不同

- cowsay All is not butter that comes from the cow.
- cowsay Hello World!
- cowsay -e ^^ Hello World!
- cowsay -f tux Tux is Linuex\'s mascot!

- man - manual (q - quit)
- man cowsay
- ls -l（“d” directory “-” file）

- apropos working directory -find commands relevant to particular keywords

- ping 8.8.8.8
- bc - calculator

- less - display text one page at a time (D, Space, arrow keys - 翻页;U 上翻页;> 最后一页)
- line number-跳转到某一行
- / search ; n 下一个；N上一个)

- nano - text editor

- pwd - print working directory

- cd - change directory
- cd ..
- cd / (回到root目录；abusolute path) relative path（从working directory开始）
- cd ~ (回到home directory)
- cd (回到home directory)

- mv - move (or rename)
- cp - copy
- mv junk trash
  如果有trash目录 move junk to trash目录
  如果没有trash目录 rename junk to trash

- mkdir - create new directories
- rmdir - remove directories
- rm -r junk 删除非空的junk目录

-man glob（globbing）

- ls *s
- ls app.{css,html} (花括号代表寻找其中任一个)
- ls a?c
- ls be[aeiou]r.png (方括号代表其中任何一个字母) beer bear

- diff -u old.html new.html

参考资料

Udacity
4.3 命令行入门@iomooc

Git学习笔记

Lesson 1

How did viewing a diff between two versions of a file help you see the bug that was introduced?

不用费劲的去比较每一行，而是可以轻松的看到更改的地方，立马让自己回忆起来自己的每一次update。
让发现bug的过程变得简单了很多。
以前居然不知道有这个功能，常常不知道两个文件内容是不是一样的，也不知道哪个文件是更新的版本，更新了哪些地方，因为一个个比较太费劲了。
现在可以让这些费劲的工作让计算机自动帮我完成，我只要根据结果进行判断就ok了，繁琐的工作交给计算机自动化，我自己进行决策就ok。这就是自动化的好处。
以前觉得决策难，是因为要获得决策的数据，和进行决策，这其实是两个部分。
就好像之前一边写文章一边还要排版一样。当我把这两个角色分开，世界就变得更清楚了。
回想起来我日常工作中觉得很累很头疼，应该也是因为有很多角色很多功能混在一起的缘故。
我觉得计算机的练习一个是可以让我学会自动化，一个是可以让我有一双眼睛能够把复杂的事情逐个分解成更简单的东西，不是混为一谈，而是条理清晰，这样解决起来就变得更容易了。
我现在觉得写reflect很有好处，我也逐渐感觉到了计算机学习对思维的训练，因为它真的可以让我看到以前没想到的维度，用我以前没用过的方式去解决问题，比如diff，比如git。
我现在真的觉得，学习python其实并不是为了学习python这种语言去编程，而是去练习一种新的看待问题和解决问题的方式。不是混为一谈，而是一点点的去拆分，分解成更简单的。

How could having easy access to the entire history of a file make you a more efficient programmer in the long term?

我可以看到我是怎么一步步改进的，看到自己的成长，看到自己从前的局限，也可以回忆起自己好的想法。
这是一种成长的印记，记录了自己改进的每一步，也容易唤起自己的记忆，知道自己是怎么走过来的。
我可以快速的找到自己更改的地方在哪里，版本更新了什么内容。
不用担心某些东西消失了，某个版本被误删了，找不到最近版本了，漏掉了之前版本里的好的内容。
不用一遍遍的保存版本1234了。更容易修改了。

What do you think are the pros and cons of manually choosing when to create a

commit, like you do in Git, vs having versions automatically saved, like Google docs does?

好处： 每一次的commit都更有意义，都是一个独立的可运行的作品。
而不是一些无意义的半成品。回顾起来更清晰。
每一个版本更新也更有意义。 坏处： 有时候可能写了很多东西，但是不记得保存了。
需要自己记得要进行手工保存。

Why do you think some version control systems, like Git, allow saving multiple

files in one commit, while others, like Google Docs, treat each file separately?

因为git中的文件常常是相互关联的，一个地方改变，可能引起其他文件的变化；或者一个改变，需要几个文件的配合才能实现。这几个文件是一个联系的整体。
而google doc中的文件基本都是独立的

How can you use the commands git log and git diff to view the history of files?

用git log看自己的更新的不同的版本，注意commit的提示
找到对应的id后，使用git diff作为对比

How might using version control make you more confident to make changes that could break something?

知道自己随时都可以方便的恢复到以前的状态，方便的追溯到问题在哪， 知道自己无论做出多么大的改变，有多么大的错误，都不会有不会挽回的后果。
有一个安全网！我就可以更自由大胆的玩耍，试验，测试。 不会再害怕冒险。
可以去尝试非常大的改变。
我觉得这样一个自由探索的空间，无论是programming，还是生活中，都是我所渴望的。

Now that you have your workspace set up, what do you want to try using Git for?

写书，包括可以进行大的改版
写文章，写作业，写心得
写小程序

Lesson 2

What happens when you initialize a repository? Why do you need to do it?

创建了一个.git文件夹，记录版本信息的变化
如果没有init的话，就无法追踪版本，只是一个普通文件夹，无法使用git的功能

How is the staging area different from the working directory and the repository? What value do you think it offers?

可以把所有文件都放在working directory里面，可以去修改，
但是在上传的时候，可以有选择性的上传几个文件，而不是所有文件。
这样可以保证我能更好地做到one commit per logical change 我可以把握修改成熟的文件上传，或者逻辑相似的文件上传，
同时继续在working directory修改。
这能搞保证我每次都可以做一个接近展示的小成品，而不是一堆草稿。
比如写文章，我可以发布我写得成熟的，或者某几个重大改变，但是其他的草稿我也可以方便的在working directory里面写下来。

How can you use the staging area to make sure you have one commit per logical change?

把这次commit需要的文件加入 staging area
不需要的文件移除staging area
同时使用git diff 确保是这次需要的添加
给了一个可以随时修改的缓冲区
这样在推送前可以有更多考虑

What are some situations when branches would be helpful in keeping your history organized? How would branches help?

试验一些新的功能 做一些探索 测试不同的方向 风格
可以让不同的版本独立发展 又可以容易的共用代码 修改代码
master相当于正式发布版保持稳定
branch可以有很多探索

How do the diagrams help you visualize the branch structure?

语言描述会比较复杂，要动用比喻，有很多分岔，不好描述
但是画图 就会感觉非常直观，会理解是怎么回溯的，一个分支为什么到不了另一个分支
所以有些概念 还是借助图
我决定在我的笔记也放些图

What is the result of merging two branches together? Why do we represent it in the diagram the way we do?

合并两个branch的功能 两个branch的commit按时排序
图的方式更直观

What are the pros and cons of Git’s automatic merging vs. always doing merges manually?

自动的优点是方便快捷 对协作 或者同时开发几个方面很有帮助
缺点是可能有冲突或者冗余
手动当然更优美 结构更清晰
但是确实也更费时 尤其是多人协作时

Lesson 3

What happens when you initialize a repository? Why do you need to do it?

创建了一个.git文件夹，记录版本信息的变化

如果没有init的话，就无法追踪版本，只是一个普通文件夹，无法使用git的功能

How is the staging area different from the working directory and the repository? What value do you think it offers?

可以把所有文件都放在working directory里面，可以去修改，
但是在上传的时候，可以有选择性的上传几个文件，而不是所有文件。

这样可以保证我能更好地做到one commit per logical change
我可以把握修改成熟的文件上传，或者逻辑相似的文件上传，
同时继续在working directory修改。
这能搞保证我每次都可以做一个接近展示的小成品，而不是一堆草稿。
比如写文章，我可以发布我写得成熟的，或者某几个重大改变，但是其他的草稿我也可以方便的在working directory里面写下来。

How can you use the staging area to make sure you have one commit per logical change?

把这次commit需要的文件加入 staging area
不需要的文件移除staging area
同时使用git diff 确保是这次需要的添加
给了一个可以随时修改的缓冲区
这样在推送前可以有更多考虑

What are some situations when branches would be helpful in keeping your history organized? How would branches help?

试验一些新的功能 做一些探索 测试不同的方向 风格
可以让不同的版本独立发展 又可以容易的共用代码 修改代码
master相当于正式发布版保持稳定
branch可以有很多探索

How do the diagrams help you visualize the branch structure?

语言描述会比较复杂，要动用比喻，有很多分岔，不好描述
但是画图 就会感觉非常直观，会理解是怎么回溯的，一个分支为什么到不了另一个分支
所以有些概念 还是借助图
我决定在我的笔记也放些图

What is the result of merging two branches together? Why do we represent it in the diagram the way we do?

合并两个branch的功能 两个branch的commit按时排序
图的方式更直观

What are the pros and cons of Git's automatic merging vs. always doing merges manually?

自动的优点是方便快捷 对协作 或者同时开发几个方面很有帮助
缺点是可能有冲突或者冗余
手动当然更优美 结构更清晰
但是确实也更费时 尤其是多人协作时

Git安装与工作环境配置

安装Git到Mac上

1.下载

Git下载

2.安装

打开其中.pkg后缀的文件，自动完成安装

3.验证安装

1. 打开Mac的Terminal。
   点击屏幕右上角的搜索图标，输入Terminal。
2. 现在可以输入git的各种命令啦。
   比如输入"git --version",
   如果已经安装成功，会显示正在使用的Git的版本信息。

Git工作环境配置

Setting Up Your Workspace on Mac

Git操作笔记

Git Cheat Sheat

Git Cheat Sheat pdf下载

Q & A

什么是Repository

一些互相关联的文件群。

为什么需要Repository

因为有时候一处修改会影响几个地方。
有时候功能的实现需要几个文件共同完成。
所以git的推送是将这些文件群作为一个整体来推送的。

git log是什么

修改日志。
里面可以看到每次修改的详细情形，比如描述，ID，更改的地方等。

练习步骤

复制Repository

1. 在Terminal中，输入"git clone 你想要复制的的Repository的URL",
   它会复制这个Repository的内容在你现有的工作目录下，
2. 如果你不知道你现在的工作目录是哪里，可以输入"pwd"
3. 如果你想更改自己的工作目录，可以使用"cd 你想使用的文件夹路径"
4. 如果想更好地使用Terminal，建议先学习一些command line的基础知识.

查看git log

1. 使用cd命令，进入你想查看的某个git项目的目录中。可以用pwd命令检验是否到达了对

的目录。
2. 输入"git log"。就可以查看到具体的git log啦。
3. 如果想退出git log的浏览，输入q（代表quit，退出）。

使用**git diff**比较两个版本的不同

1. 根据git log，找到自己想要比较哪两个版本。
2. 复制commit后的一长串字母和数字组合的id。
3. 输入“git diff 第一个id 第二个id”进行比较。
4. 如果觉得id太长，也可以只输入id的前面4个字符。
5. 如果想退出，输入q。

回到上一个版本的状态

1. 输入"git checkout 想回到的版本的id"

Gitbook与Github互推

觉得Gitbook是个好东西，顿时有种想写书的感觉啦。
只是之前以为gitbook和github是自动同步的，因为看上去都是git家族的嘛>_<
结果没想到要实现这个还有自己设置，而且居然一下子还木有设置成功，这个问题就一直闲置在那里几天了。
结果没想到今天试一下居然成功了，记录自己的体会。

解决问题的启示

为什么第一次没解决

- 第一次有发现import tool，提示要输入git url， 其实是要输入book的git url，结果我以为是输入github的url，一直不成功。
- 看到英文界面，加上对github完全不熟悉，有种无从下手的感觉。
- help文档其实很容易看到，但是一直被我无视。

为什么今天尝试成功了

- 学习了一点github知识，对这个没有那么怕了。
- 正好看到有help文档，发现其实有提到这个，就跟着边看边做。

启示

- 学会使用帮助文档。
- 对一个知识不熟悉而害怕，可以先补一下入门知识。
- 当时解决不了没关系，过几天学了一些别的，回过头来可能忽然就会解决了。

> 参考资料：Gitbook帮助文档

步骤：

从Gitbook传送到Github

Transferring content to GitHub

- Use the Import Tool from GitHub
- Enter your GitBook git url,

for example: https://git.gitbook.com/MyName/MyBook.git
(this url can be found in your book settings).
打开网站右上角的“MY BOOKS”，点击你想传送的书名。
打开后，选择右下角的“Settings”。
页面往下翻，可以看到有个Git URL，就是这里要用的地址。

- Enter a name for your GitHub repository.
- Enter your GitBook credentials (you can use your API token instead of your password) when prompted.

连接Gitbook与Github账户

GitHub Integration

When your content as been transferred to GitHub, you can now setup the integration so that GitBook can still build your book from GitHub。

1. Connect your account / Permissions

In your account settings, connect your GitHub account with the correct permissions:

- 点击网站右上角自己的头像，选择Account Setting.
- 页面往下翻，看到有个Github设置的地方，设置自己的Github账户信息。
- 注意这里有个三角形的下拉选框，选择Access to public repositories。

2.Link a book and a GitHub repository

Open the GitHub section in your book settings Enter your repository id (such as: YourGitHubUserName/RepoName) Save your settings

- 打开网站右上角的“MY BOOKS”，点击你想传送的书名。
- 打开后，选择右下角的“Settings”。
- 进入Setting后，点击右边菜单栏最下面的Github。
- 你可以看到设置你GitHub Repository的地方。
- 输入你的repository id (如: YourGitHubUserName/RepoName)
- 保存设置。

3.测试

- 在gitbook里对图书修改，看是否同步到了github。
- 在github里对图书修改，看是否成功同步到了gitbook。
- 如果有问题可以对照帮助文档，继续解决。

为Gitbook添加评论-disqus

思考 - 想为gitbook添加评论

- 想到评论就想到之前有在自己的blog上使用过disqus的插件。
- 很多网站也使用了disqus。
- 看到别人的gitbook里有使用disqus的。
- 初步决定使用disqus。或者再搜索一下有没有好的comment pluggin。

摸索

- gitbook的右上角的Pluggin里搜索，有点看不懂。
- google搜索，好像大家都是在本地安装的gitbook，在本地配置的。我想在网站上直接配置。
- 不管了，跟着来，能走一步走一步，先看效果。
- 居然就可以了，明明很多步骤还没做的说，原来只做其中的几步也可以成啊。

启示

- 可以先看看别人做到的人时怎么做的，给自己一些启发。
- 先做自己能做到的，即使还有很多步骤没做也没关系，多走一步又会看见新的风景。

步骤

如果希望别人能够在自己发布到GitBook平台的电子书文章下评论，可以使用第三方评论工具Disqus

注册disqus

- 打开:https://disqus.com/
- 点击"Sign Up",输入邮箱,用户名,密码,创建账号
- 新建站点
- 登录后,点击右上角的齿轮,选择"Add Disqus To Site",填写"Site name",这是book.json里面要填的"shortName","Finish registration".

备注：
"shortName"和注册时填写的用户名作用不同: 用户名是账号的名称,而"shortName"是这个新建站点唯一的名称.一个账号可以有多个站点,也就有多个"shortName".

在网页上使用这个"shortName"以后,这个网页下面的所有评论都会发送到disqus上对应"shortName"的站点. 打开网页时,"shortName"告诉disqus需要加载并显示哪个站点的评论.

插件安装

- Gitbook中，在要编辑的书里， 点击右上角的下拉三角形，选择Edit Book Configuration.
- 系统会自动生成book.jason
- 添加下面的内容。

```
{
"plugins": ["disqus"],
“pluginsConfig”: {
"disqus": {
 “shortName":” “NAME-FROM-DISQUS”
  }
}
}
```

- 将NAME-FROM-DISQUS 改为上面提到注册时写的shortname。

参考链接

首次试

~ 此目录收集, 入门时,编程的各种冲突

进展

- 150924 大妈创建

Week2:直面生活

Zoe:
直到今天看到弓和箭给六个月前的自己写信的时候， 我才想起，其实我也该为你写一封信。

其实这个教程是写个你的， 只是我有时候会忘记了。
有的时候， 我以为我是要写一个给别人看的教程， 有时候我又以为我是要写一些知识的笔记，
有的时候我又觉得自己是在写日记而已。

其实，六个月前的你在乎这些吗？不在乎。
是的，不在乎。

知识忘记了又怎样，google一下，答案总会有。
编程又怎样，如果只是把它看做一门技能，学了又怎样？
写日记又怎样，写过很多的日记，可是还是迷茫，还是难过，还是找不到出口。

这就是为什么你之前会不停的开始，又不停的放弃。
因为，你觉得没有意义。没有意义。

我知道，你的内心里，是隐隐期盼着一个钥匙，打开新世界大门的钥匙。
你希望，在编程的世界里，可以让你学会推开另一个世界的大门。
可以终于学会解决那些困扰过你的问题，那些困扰着你的生活。

所以我知道，光说编程没法打动你，要给你一把钥匙，打开新世界的大门，感受新世界的思维。
所以我知道，编程不能光只是好玩的虚拟世界的小打小闹，你希望它能给你的现实世界带来真实的改变。

亲爱的，我知道你很害怕，害怕生活会就这样一直下去，
害怕在每个半夜醒来的夜里睡不着，
因为你知道，你的内心并不喜欢这样的生活，你无法再欺骗自己。

很高兴有这样的三个月的时光， 给你写信，一起讨论一些你真正关心的事情，
一起真正的去面对这些问题，并学着用新的思维去解决这些问题。
这一次，不再是心灵疗伤或者是什么心理安慰，
而是，真正去解决这些问题。
想一想，这是不是很酷？

这也是这十多天捣鼓编程给我带来的新的改变，
发现问题，折腾，尝试，搜索，请教，
想尽办法的去解决问题。

你觉得开心吗？
你是不是该觉得，Zoe呀，你可终于开始长大啦？

是啊，小Zoe，我们会长大，用我们喜欢的方式。

你不是说过，你好想要有一个可以让自己自由成长的环境吗？
就让我们一起，学着一起创造一个这样的环境，创造一个真正的Zoe会喜欢的环境。
就让我们在现实的世界里，给自己一份自由呼吸和成长的天地。

编程不是你用来逃避世界所培养的爱好，
而是可以成为你面对生活的有力伙伴。

不要害怕，不要害怕。
其实21世纪的你，拥有很多很棒的资源和能力，真的。
我还知道，你的内心里有一团火焰，不想熄灭，虽然它已经快奄奄一息，但是它还是想要表达出来的，不是吗？

我知道，你不想成为一个无聊的职员。
我知道，你看到他们的生活，觉得自己的未来竟然就像其中的某一种，觉得好可怕，真的好可怕。
你想抱头大哭。

Zoe，我想和你说，其实你有选择的，知道吗？
或许在他们的时代，没有这么多选择的机会，
那个时代没有互联网，没有计算机，没有英语，难以看到外面更大的世界，而现在的你可以。

暂且不说别人，他们变不变也不是我们要操心的事。
我就想说说现在的你，
你想要改变，你觉得不改变很痛苦，
你所处的时代也赋予了你很多可能性，
你也拥有一定的能力去改变，
这样就足够了。

看看是不是真的那么难。
小Zoe，我和你打个赌好啦，我觉得只要你去行动，改变没有你想的那么难。
你一定又要说我站着说话不腰疼了。
好啦，这一切当然也不容易，我也猜到你会碰到很多艰难险阻，不过比起你每天这样要死不活愁眉苦脸的可要好多啦。
你不是还绝望的觉得这辈子就这样了？
哼，我才不信呢。

没关系，反正我们今天就要开始直面生活，用编程思维改变生活了。
我们也不小打小闹，培养什么兴趣爱好，搞什么副业什么的，我们就想真的去改变，
这三个月，让我们好好的学着去改变，去行动，去解决问题，去靠近我们更喜欢的生活。

Zoe Baby，加油！
Zoe Baby，长大成自己喜欢的样子。
Let's go!

工作中两种类型的事情

Part.1 - 是个人都能做的事情

工作中大量的事务性的重复无聊工作，让人深恶痛绝啊。

其实这种事情，随便一个中学生就可以做了，
或者，根本都不用人来做，换个机器来做效果都一样。

我们为什么要花几十年的光阴在这种事情上，数十年如一日的做这种无聊的事情？
无聊就算了，还这么容易被取代，换个人来做其实是一样的。
这样的工作经验有积累的意义吗？
我觉得这就是对自己的一种浪费。多做几年下来，人都会变傻。

正是这种无聊的没有意义的重复的繁琐的事务性工作，耗费了人大量的时间和精力，导致我们没法用更多精力去做一些更重要的事情。

想一想这些破事：

- 反复登陆某个网站几十次，等待网站缓慢的打开，退出，再登陆。
- 每天给别人发几条群发无聊短信。
- 在网站上重复的添加几十条类似的记录。
- 用本子抄写各种东西。
- 每周看各种东西够不够用，调拨。
- 手抄各种登记本，盖章，写错了重抄。
- 反复在一个表中寻找某两个数字对应的数值。
- 登陆系统打印表格，计算机加数字，打电话告诉别人。

我觉得，有些东西我可以靠编程自动化去实现，起码更智能点，
有些东西，我可以提建议去优化系统，虽然不见得马上有效，但不合理的东西终会慢慢淘汰，
有些东西，我暂时无法改变，但是我可以让我生活中无聊的事情比重少一点。

少做点是个人都能做的事情。
太容易被替代。
万一有天被辞退了，都不知道去哪里找新的工作，干这种容易的事情的人可不缺。

Part.2 - 相对需要更多积累的事情

当我觉得有些东西是计算机比较难以自动化的，我觉得要费点脑子的事情，我觉得这才是更值得去做的事情。
但是当有一大堆的事情的事情摆在面前的时候，这种事情就会变得难以识别，甚至因为有点难做而被轻易放弃了。

- 写作
- 演讲
- 主持
- 讲课，培训
- 优化系统
- 优化流程
- 系统的学习

让编程成为我工作中的好帮手

- 能自动完成的事情，交给计算机自动完成
- 不能自动完成，起码给我更多的决策建议，比如我今天要干些什么，怎么干，我容易出错的地方在哪里。不用我在不同的笔记里跳跃来跳跃去寻找要干嘛，容易遗漏。
- 记录中工作中犯的错误，看可否从流程中优化，下次减少这种错误。
- 享受有这样一个好伙伴的快乐，让每天上班不烦躁，心情舒服。
- 花更多时间去做些更让自己开心更充实的事情。

工作问题解决思路

- python
  python自动化的书，最符合我的思路，加上又是python的，值得好好读。
- AutoHotKey
  曾用它实现过类似的功能，可以继续学习。
- Excel
  数据的处理，是不是用excel更方便一点，这个也可以考虑

提醒

时间精力有限，注意**Focus**

不要想着什么都要做，
现阶段先看python那本书，学点编程思维再说。
先把python能解决的事情解决再说。

学会使用别人造的轮子

真正解决问题，不用从零开始，
学会用好这个世界上优秀的人聪明的人提供的好工具好代码好课程。
学会用轮子，就会发现很多问题的解决很容易，而且效果很好。

学会请教和求助

喵的，付费参加这个课程，请好好利用资源，不要浪费！

编程思路探讨—— 小游戏：guess the number

游戏描述-猜数字

计算机先想出一个1-20之间的数字，玩家可以猜6次，看能否猜中数字。

思路一

分析input和output

input

- 游戏者的名字
  - name=input()
- 游戏者猜的数字
  - guess=input()

output

- 向游戏者的问候
  - Hello, what is your name?
  - Hi, xxx , let's play a game - guess the number.
- 游戏引导（介绍数字的范围）
  - Well, I am thinking a number between 1 and 20.
  - Take a guess.
- 游戏提示
  - 数字太大
    - Your guess is too high.
  - 数字太小
    - Your guess is too low.
- 游戏结束
  - 正确猜出（给予赞扬）
    - Good job! xxx, You guessed my number in x guesses!
  - 没有猜出（告知答案）

- Nope. I number I was thinking of was x.

分析progress

- 用random生成一个随机数字。 secretNumber
- 比较secretNumber和guess的大小，if...elif...
- 可以猜x次。for i in range（）

尝试写游戏

思路二

简单版本1

- 计算机想好一个数字
  - secretNumber=random.randint(1,20)
- 我猜一次
  - guess = int(input())
- 告诉我成功还是失败
  - if...print
  - else...print

简单版本2

- 我猜1次 -> 6次
- 告诉我成功还是失败 -> 告诉我是大了还是小了

简单版本3

- 优化游戏的文字部分，使得它更像个游戏

思路比较

- 思路一思考起来更直观，前期思考会更多些，更能从宏观上把握架构，写出来的东西逻辑性和整体性更强
- 思路二更能马上就开始动手写程序，并且可以马上就运行起来，每次思考过程都有一个独立的可运行的版本，然后不断去改进

我的游戏代码参考

```
import random

print('Hello, what is your name?')
name=input()
print('Hi,  ' +str(name)+ ", let's play a game - guess the number.")

print('Well, I am thinking a number between 1 and 20.')
secretNumber=random.randint(1,20)

for i in range(6):
    print('Take a guess.')
    guess=int(input())

    if guess < secretNumber:
        print('Your guess is too low.')
    elif guess > secretNumber:
        print('Your guess is too high.')
    else:
        break # guess is correct

if guess==secretNumber:
    print('Good job! '+str(name)+', you guessed my number in '+str(i+1)+' guesses!')
else:
    print('Nope. I number I was thinking of was '+str(secretNumber))
```

Running Programs from the Command Line

> 参考资料 Appendix B – Running Programs

Shebang Line

The first line of all your Python programs should be a shebang line, which tells your computer that you want Python to execute this program.

- On Windows, the shebang line is #! python3.
- On OS X, the shebang line is #! /usr/bin/env python3.

You will be able to run Python scripts from IDLE without the shebang line, but the line is needed to run them from the command line.

Running Python Programs on Windows

- create a .bat batch file
    - @py.exe C:\path\to\your\pythonScript.py %*
- Replace this path with the absolute path to your own program, and save this file with a .bat file extension.
- The MyPythonScripts folder should be added to the system path dialog.
- run the batch files in it from the Run dialog.
    - pythonScript

Running Python Programs on OS X

- Save your .py file to your home folder. (Or change the PATH)
- Then, change the .py file’s permissions to make it executable by running
    - chmod +x pythonScript.py
- you will be able to run your script whenever you want by opening a Terminal window and entering
    - ./pythonScript.py.

用python快速打开网页

> 更新日志
>
> - 20151016 第一次-第四次尝试

第一次尝试 遇见webbrower

搜索how to open web browser in python，学习到了webbrower

```
import webbrowser
webbrowser.open(url)
```

发现是是Safari打开的，我想用chrome打开

第二次尝试 使用chrome

搜索python webbrower chrome 发现 Python webbrowser.open() to open Chrome browser @stackoverflow
非常棒的解决方案，好好学习

```
import webbrowser

url = 'http://docs.python.org/'

# MacOS
chrome_path = 'open -a /Applications/Google\ Chrome.app %s'

# Windows
# chrome_path = 'C:\Program Files (x86)\Google\Chrome\Application\chrome.exe %s'

# Linux
# chrome_path = '/usr/bin/google-chrome %s'

webbrowser.get(chrome_path).open(url)
```

第三次尝试 在python外运行

参见Running Programs from the Command Line

- Add shebang line in python file
  - `#! /usr/bin/env python3`
- change the .py file's permissions in Terminal
  - `chmod +x pythonScript.py`
- run it in Terminal
  - `./pythonScript.py.`

第四次尝试 制定不同的网址（使用参数）

尝试搜索 python webbrowser parameter|argument
看不懂
试着使用sys.argv未果
不过正好链接到了我在学习的Automate the boring stuff的课程
于是决定继续学习课程

学习到了sys.argv的使用方式

以下代码实现的是
Terminal打开 `./mapit.py ADDRESS(或者复制ADDRESS到剪贴板上)`
能够自动用google map打开该地址的地图

```
#! /usr/bin/env python3
import webbrowser, sys, pyperclip

sys.argv #['mapit.py', '870','Valencia','St.']

# Check if command line arguments were passed
if len(sys.argv) >1:
    #['mapit.py',      '870','Valencia','St.'] -> '870 Valencia St.'
    address = ' '.join(sys.argv[1:])
else:
    address = pyperclip.paste()

# https://www.google.com/maps/place/<ADDRESS>

webbrowser.open('https://www.google.com/maps/place/' + address)
```

让python自动获取天气

> 更新日志：
>
> - 20151017 添加第一次-第二次尝试

第一次尝试

用beautiful soup和requests来完成，
总是提示
NotImplementedError: Only the following pseudo-classes are implemented: nth-of-type. 猜测是天气预报网站的css标签是beautiful soup不支持的，或者是我没找到合适的引用方式。

第二次尝试

搜索python get weather
发现了python-weather-api

安装完成后，获取城市株洲(id：CHXX5650)的天气：

```
import  pywapi
import string

weather_com_result = pywapi.get_weather_from_weather_com('CHXX5650')

print("Weather.com says: It is " + str.lower(weather_com_result['current_conditions']['te
```

拓展思路

- 搜索今日的预报，而不是仅仅是当下的天气
- 可以自己搜索城市名称，不输入就自动反显株洲

Week3:开始编程啦

看到自卑，看到不敢，看到就会有新的发现

——第一周视频课程笔记

> 更新日志：
>
> - 20151018 看到issue提到的5W1H，将具体做法改得更详细一些了
> - 20151018 创建

最触动我的点：自我审查

Hi，Zoe.
你知道吗？今天听大妈的视频，突然很触动我的，是关于自我审查的那一段。
不敢做，觉得自己不好，总觉得自己想要的做的事情需要获得别人的同意，总觉得自己的要求更低一等，别人的要求更有道理。看低自己，看清自己，觉得自己不如别人。
为什么要贬低自己？为什么不能把自己当成一个正常的人呢？
为什么总觉得自己是无力的螺丝钉呢？
想想自己这么多年，从很听话，到叛逆，然后再到现在的重新听话，我觉得还是有点迷失自己。
其实我突然很想念叛逆的自己，自己喜欢的事情，就可以不管不顾的去做的自己，看到一点点可能性就真的可以搞定的自己。
但是我不知道为什么，我现在很软弱，看不清自己想要的，分不清自己的声音和外界的声音，也不懂得拒绝。 为什么就是觉得自己的需求是不合理的？觉得自己喜欢的是不好的？只能偷偷去干的？

其实连学编程也是这样，如果不是付款的时候被碰到，我觉得我是不好意思和老公说，我要花钱学编程的。
要和他说要花一千多块钱干这个事情，总有点说不出口，好像亏欠了他一样。
然后想到自己之前有很多东西半途而废，真的不知道这次可不可以，虽然我很想，但是是不是自己花钱太随意了？
总觉得女孩子说喜欢学编程，是不是爱好有点怪怪的？
觉得每天总想对着电脑，是不是有点脱离真实世界？
想到自己每天不钻研工作的事情，也没有积极做家务搞饭什么的，钻研编程是不是感觉特别不务正业？



Zoe，你还记得吗？
你那时候说，总觉得有一股拉力和一股阻力，让你总是一次次开始，又一次次放弃，你说不知道这股阻力是什么。
看了上面的那一段内心的问题，你是不是逐渐能看得更清楚一些了？ 是啊，我可以说出我为什么想要学编程，但是有时候我的内心无法阻挡的，是我的另外的没有说出口的问题。
我害怕，我不敢，我觉得我不对，我觉得我不应该。

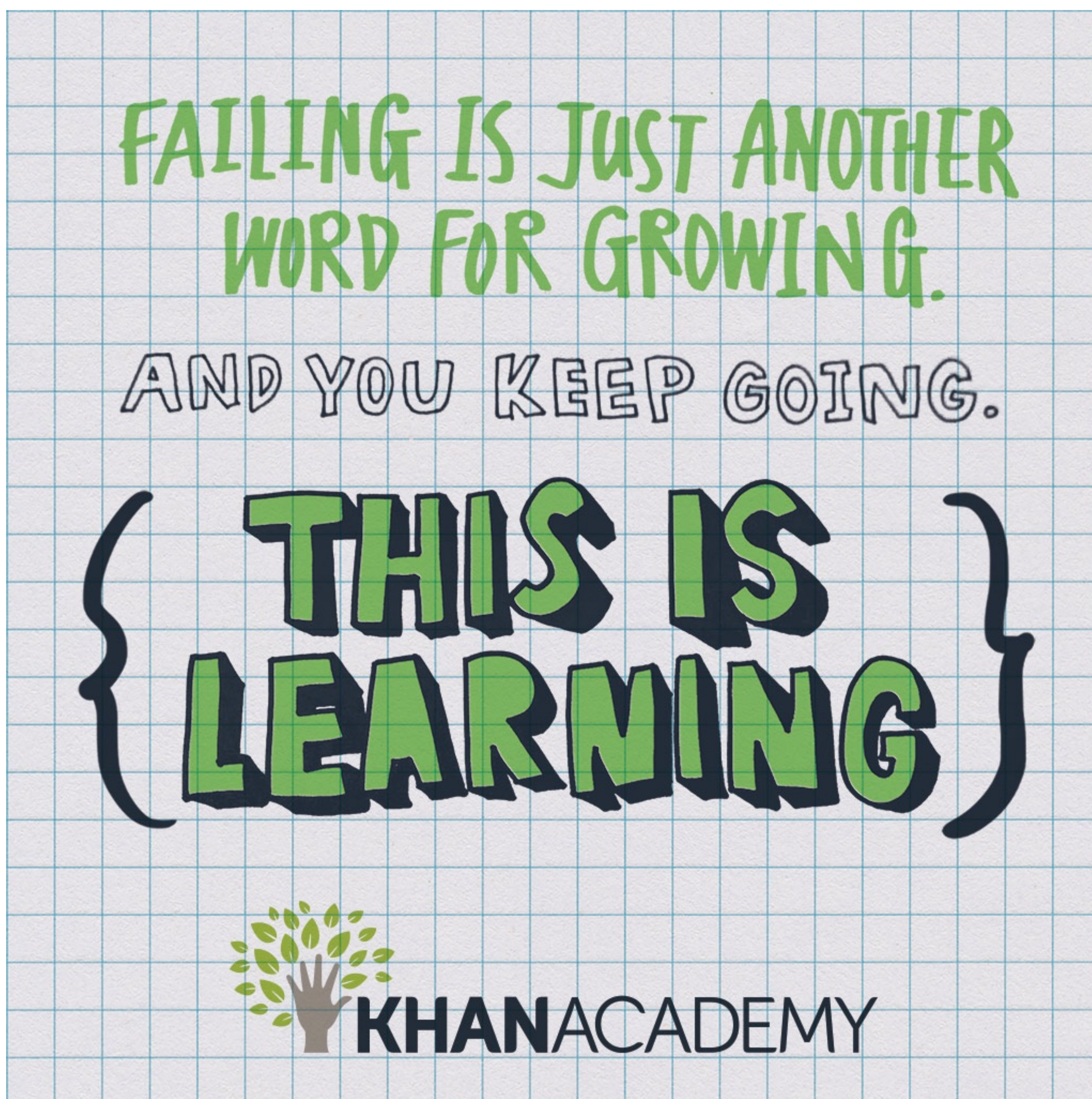
然后想起大妈说的，为什么要贬低自己呢？为什么不能把自己当成一个正常人呢？
是啊，就像一个正常人一样，有开心，有失意，有成功，有失败，有自己会的，也有自己不会的。

我觉得在工作之后，有点找不到自己的节奏了，
像Pink Floyd唱的，总觉得自己是Another Brick in the Wall. But, is this true? 大人们说，你就是一颗螺丝钉，企业少了谁，一样都能转。
但是，我们真的要用这样的方式去看待自己吗？
这样过一生，有什么意思呢？为了演一出别人其实也不在乎的戏吗？

想起小时候，就是喜欢探索，就是觉得好玩，就是觉得来劲，
不是为了证明自己的优秀，不是为了证明自己有能力，不是为了某一天变得牛逼，不是为了和别人比来比去。
我喜欢，我想投入，我想做，我想玩，就是这么简单。

Zoe，我知道你会害怕。没关系，来到自己不熟悉的地方，来到自己模式的领域，都会有些害怕。
还记得第一次在ToasterMaster发言的时候，自己一直都在抖抖抖吗？然后到后来，看到自己原来也可以做到，原来没有想象中的难。
是的，我们需要尝试自己不熟悉的事物，就像《学习的艺术》说到的两种棋手，固定心态拒绝长大，还是成长心态去面对新的挑战。
我们总会要去面对自己原本不熟悉的事物，但也正是因为这样，我们可以成长到，能做到以前自己做不到的事情的人。

还记得Khan学院里的那个视频，You can learn anything吗？记得每个人，都是从最初跌跌撞撞开始，然后慢慢成长的吗？
有的时候，我们会战战兢兢，觉得自己不行，就像Khan的小侄女一样，战战兢兢的想要放弃。
但是我知道，其实你也想遇到一个像Khan一样的人，去鼓励自己，去认真的教自己，去慢慢的补上自己之前的漏洞，然后好好的成长起来。
其实，互联网就给了你这样的机会，你也应该给自己一个这样的机会，不是吗？



Zoe，其实你最喜欢的，就是一片可以自由学习和成长的环境，同时也是充满鼓励的，能够让你更听清楚内心声音的环境。
Zoe你知道吗，你已经长大了，你也拥有很多能力，为自己创作一个这样的环境啦。
对自己温柔一点点，耐心一点点，相信自己多一点点，不要那样轻易的贬低自己，放弃自己，好吗？因为这样，你自己会伤心的。
因为你知道，其实你最需要自己的认可和鼓励，需要内心对自己的相信。
一开始可能并不容易，可能会摔跤，可能会沮丧，可能会碰壁，可能会失败，但是，让我们学着开始，好吗？
从今天开始，学会对自己温柔一些，从今天开始，学会自信的说出自己的喜欢，说出自己的需求。

触动自己的top3

话说今天好像有点偏题了，大妈不是说要写最触动自己的top3吗？

- 一、更温柔的对待自己，更自信的说出自己的喜欢。不要轻易的贬低自己。想一想自己小时候，想遇到一个怎样的自己？学着用自己喜欢的方式对自己。
- 二、学编程是为了学习编程思维，抓住主要矛盾解决问题。我们时间精力有限，不可能面面俱到追求完美，能够抓住主要矛盾，能够解决问题就OK。而且新的思维和视角，可以让我看到新的可能性。
- 三、墙只存在于心里，真的像翻越，方法太多了。就像Randy说的，墙在这里，只是为了证明我有多想要。墙能拦住别人，但是拦不住我。

如何解决我前面说到的这些问题呢？

- 一、抓住主要矛盾，不可能面面俱到，不可能尽善尽美。
  学编程是学什么？其实不是具体的编程语言和技巧，不是写个能运行的程序。 而是新的思维方式，我完全没有想到的地方，这些思维点，需要细细钻研。
  所以，今后的学习重点是，我没有看到什么，我对什么视而不见，什么是我没有意识到的，什么是完全没有想过的？
  现在我知道了这个，可以对我有些什么新的改变，我可以马上运用起来吗？有些事情我会换种方式去解决吗？
  这次三个月的入门班，不可能什么都一把抓，重点放在思维扫盲上。

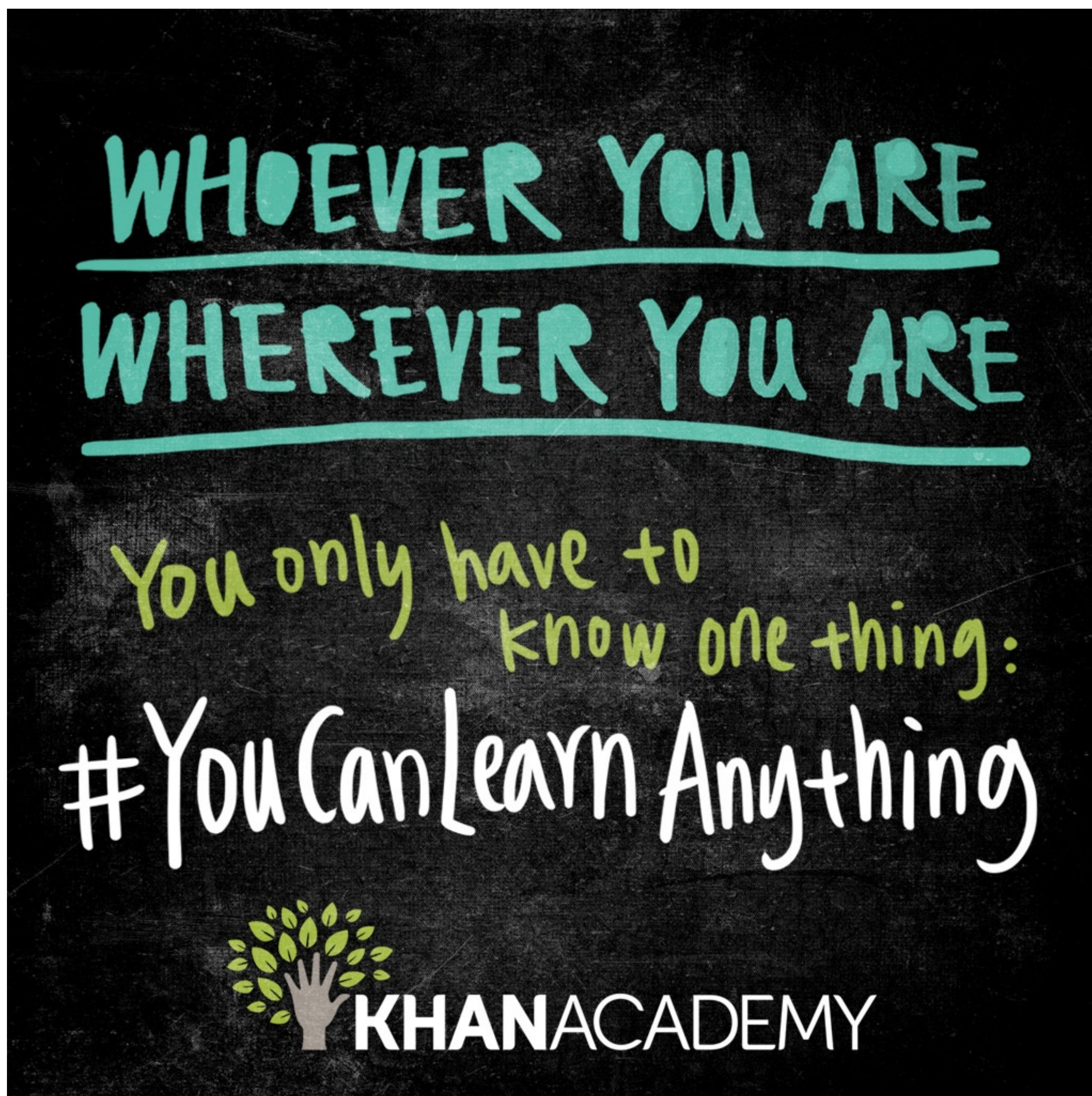
- 二、真的想翻越一堵内心的墙，我要做些什么？
  一个是上面提到的全新的思维方式，之前的自己完全没有想到的可能性，完全没有看到的可能性。 尝试，尝试是不是真的那样难以逾越，行为上做些以前不会做的事情。
  借助peer pressure，是的，peer pressure真的力量很大，让自己融入到一个新的圈子，为自己创造一个全新的圈子，无论是人，还是信息来源，这会对自己有新的改变。

- 三、从自我审查到自己鼓励，尝试，靠近。
  写下自己的担忧，让看不见的担忧变得看得见，然后写下自己新的对话。
  不是空想或者自我安慰，而是用行动去靠近。

具体做法：

1. 每周开始时，记录自己的Weekly Goal@Evernote。 每天学习前，根据今天的学习时间，先写下自己今天要完成的一个目标@Evernote。 记录我这周的重点和主要矛盾，我真正想做的事情。提醒自己把注意力放过来。提醒自己不可能面面俱到。不断修正。
2. 每天学习后的Review@Evernote，除了之前的写下自己干了什么意外，也要记录之前自己完全没想到的想法，没看到的地方，可以在自己的思维盲区花点时间，重点突破。
   Gitbook上每次更新的教程，不仅是写出正确步骤，更要写下为什么会犯错，以后怎么避免类似错误。
3. 心情不佳的时候，在evernote里写下自己的担忧，看清楚自己的担忧，同时想一想小时候的自己希望能够遇到一个什么样的伙伴，和自己构建新的对话。

4. 构建自己更喜欢的交往圈和信息源。认真写作，认真分享，认真回复。靠近自己内心自然喜欢的人。学会删除无用的信息源，学会看英文信息源，从学习看帮助文档开始，从github接触创作者开始。



交互101 - 小小日记系统

> 更新日志
> 20151018 创建，进行思考一和尝试一-三

卡片一：本周整体任务概述:

- 完成一个极简交互式日记系统，需求如下：
    - 一次接收输入一行日记
    - 保存为本地文件
    - 再次运行系统时,能打印出过往的所有日记
- 时限: 0wd4~1wd3
- 发布: 发布到各自仓库的 _src/om2py0w/0wex1/ 目录中
- 指标:
    - 包含软件使用说明书: README.md
    - 能令其它学员根据说明书,运行系统,完成所有功能

思考一：

- 输入“diary.py 日记内容”，将内容保存为本地文件
- 要用到import sys的sys.argv
    - if len(sys.argv) >1：（含有日记内容）
        - 输入一行文字到diary.txt
        - 如果diary.txt不存在，创建它
- 输入‘diary.py’，打印以往所有日记
    - if len(sys.argv) >1：
        - 打开diary.txt
        - 如果diary.txt不存在，创建它

Question

- 如何用python创建diary.txt
- 如何用python添加文字到某一个文档

Solution

- 学习 Lesson31 Reading and Writing Plaintext Files@Automate
    - append mode就是我想要的

- 可以在文末增加一行
- 如果文件不存在也会自动创建文件

```
helloFile = open('hello.txt','a')
helloFile.write('Hello')
helloFile.close()
```

尝试一：

```
#! /usr/bin/env python3

import sys


if len(sys.argv) >1:
    diary=open('diary.txt','a')
    diary.write("\n" + content)

else:
    content = diary.read()
    print(content)

diary.close()
```

之前有担心不能输入中文，后来发现顺利通过。

Question

- 如何判断是不是第一次添加日记（日记是空白），这时不用添加空行
- 发现直接调用d.py的时候，可以添加中文没问题，
  但是直接编译的话其实是会报错的，
  估计还是diary.txt里有中文的问题，
  先记录，待解决。

```
Traceback (most recent call last):
  File "/Users/Zoe/GitHub/MyPythonScripts/d.py", line 14, in <module>
    content = diary.read()
  File "/Library/Frameworks/Python.framework/Versions/3.5/lib/python3.5/encodings/ascii.p
    return codecs.ascii_decode(input, self.errors)[0]
UnicodeDecodeError: 'ascii' codec can't decode byte 0xe6 in position 1: ordinal not in ra
```

尝试二：

Quesiton

- 如何添加日期

Solution

- Python: Get Today’s Current Date and Time

```
import time
print(time.strftime("%Y/%m/%d"))
```

代码

```
#! /usr/bin/env python3

import sys
import time

diary=open('diary.txt','a')

if len(sys.argv) >1:
    content = ' '.join(sys.argv[1:])
    diary.write("\n" + time.strftime("%Y/%m/%d")+ ' ' +content)

else:
    diary=open('diary.txt')
    content = diary.read()
    print(content)

diary.close()
```

尝试三：

- 试着发布，就会觉得readme写的好烂，今晚先跌跌撞撞这么写着吧，以后要加强
- 当要发布给别人用的时候
  - 发现不仅要考虑自己在用的Mac OS下的Python3，还要考虑python2，还要考虑Windows
  - 发现还要告诉他们将环境配置成和我类似的，比如Path，chmod这些

Question

- 可不可以用类似setup.py的功能，自动完成这些配置工作

思考二

看issue回复的时候，觉得是不是还是在py里面进行互动和输入会更好一些？
这样界面会更友好，提示性也更强一些，
不会出现我自己过了好久都不记得怎么用的情况，
别人用起来也觉得一目了然，不用去翻帮助文档。

同时，互动文字可以让程序更有爱一些。
感觉好像有一个人在和你对话，而不是硬邦邦的代码。
记得看那本和game有关的python的书时，最打动我的就是它的举例的程序，运用的互动语言都很有感情，让我觉得编程原来也可以是这么有感情的事。

- 输入diary.py
- 出现提示：
    - Hi. I am the diary of yours.
    - Do you have something want to share with me now?
    - Or you can just press Enter to read all of me.
- open file 'diary' , append mode
- newDiary=input()
- 出现提示：
    - Thank you for sharing with me.
    - Bye. I will miss you.
- close file

尝试四：

感觉大家还是使用python2的比较多。
所以这次编程准备在python2.7的环境下编写。

Question

- input() not working in python2
- 没发现中文有什么问题呀，大妈说的中文输入问题到底指的是什么呢？

Solution

> Use raw_input() instead of input():
> `testVar = raw_input("Ask user for something.")`
> input() actually evaluates the input as Python code.
> I suggest to never use it.
> raw_input() returns the verbatim string entered by the user.
>
> Although for anyone reading this using Python 3, input now works this way, and raw_input is gone.
>
> via Python 2.7 getting user input and manipulating as string without quotations

思考三

再次受到issue的启发，
包括那时候上Kunal的课，其实最感动的是，当他们用python写出一些很简单但充满爱的小程序，和他们所爱的人一起分享的时刻。
无论是给自己的小孩子上一课超简单的python课，还是用python做出一个小小照片游戏，告诉自己亲爱的人，说爱你。
我想，这是我觉得python最有爱的时刻，也是最最让我感动的时刻。
这程序

让编程更有爱的一点 - 继续小小日记

> 更新日志
> 20151024 增加哥哥写的日记
> 20151020 创建

小小日记任务回顾

- 完成一个极简交互式日记系统，需求如下：
    - 一次接收输入一行日记
    - 保存为本地文件
    - 再次运行系统时,能打印出过往的所有日记
- 第一次笔记
    - 小小日记系统-第一次尝试

思考二

看issue回复的时候，觉得是不是还是在py里面进行互动和输入会更好一些？
这样界面会更友好，提示性也更强一些，
不会出现我自己过了好久都不记得怎么用的情况，
别人用起来也觉得一目了然，不用去翻帮助文档。

同时，互动文字可以让程序更有爱一些。
感觉好像有一个人在和你对话，而不是硬邦邦的代码。
记得看那本和game有关的python的书时，最打动我的就是它的举例的程序，运用的互动语言都很有感情，让我觉得编程原来也可以是这么有感情的事。

- 输入diary.py
- 出现提示：
    - Hi. I am the diary of yours.
    - Do you have something want to share with me now?
    - Or you can just press Enter to read all of me.
- open file 'diary' , append mode
- newDiary=input()
- 出现提示：
    - Thank you for sharing with me.
    - Bye. I will miss you.
- close file

尝试四：

感觉大家还是使用python2的比较多。
所以这次编程准备在python2.7的环境下编写。

Question

- input() not working in python2
- 没发现中文有什么问题呀，大妈说的中文输入问题到底指的是什么呢？

Solution

> Use raw_input() instead of input():
> `testVar = raw_input("Ask user for something.")`
> input() actually evaluates the input as Python code.
> I suggest to never use it.
> raw_input() returns the verbatim string entered by the user.
>
> Although for anyone reading this using Python 3, input now works this way, and raw_input is gone.
>
> via Python 2.7 getting user input and manipulating as string without quotations

代码

```
import time

print('Hi, I am the diary of yours.')
print('Do you have something want to share with me now?')
print('Or you can just press Enter to read all of me.')

diary = raw_input('>>')

if diary == '':
    diaryFile = open('diary.txt')
    diary = diaryFile.read()
    print(diary)

else:
    diaryFile = open('diary.txt','a')
    diaryFile.write('\n' + time.strftime('%Y/%m/%d')+ ' ' +diary)

    print('Thank you for sharing with me.')
    print('Bye. I will miss you.')

diaryFile.close()
```

思考三

再次受到issue的启发，
包括那时候上Kunal的课，其实最感动的是，当他们用python写出一些很简单但充满爱的小程序，和他们所爱的人一起分享的时刻。
无论是给自己的小孩子上一课超简单的python课，还是用python做出一个小小照片游戏，告诉自己亲爱的人，说爱你。
我想，这是我觉得python最有爱的时刻，也是最最让我感动的时刻。
那些程序是那样的简单，却让我真的感觉到，这样简单的事情是如此的有意义。

我想，为什么要把编程这件事搞得好像那么深奥和复杂呢，好像难以和不了解编程的人沟通呢？
想起之前都是自己在这里看视频，学写代码，感觉好像也不知道怎么和老公说这件事情一样，为什么不用一种别的形式来一起分享这个过程呢？
包括他问起来我写了些什么，我好像一下子也难以描述清楚。
为什么不把这件事情从自娱自乐，转变成和他一起来分享呢？
虽然我自己也觉得好玩啦，不过是不是之前根本没这么想过？
是这件事情很难，还是我根本没有尝试过呢？

失败也没有关系啦，起码我也可以看看他的反馈，自己再改进嘛。
而且我觉得这样应该还蛮好玩的。

喵，好高兴今天突然想起了那时候Kunal传达给我的这种充满爱的感觉。
让我感觉，编程就是生活的一部分，编程让生命更有爱。
编程不仅仅是关于计算机，更是关于人，关于我所爱的人。

About People.
About Love.
About Life.

Zoe, remember this feeling.

尝试五 - Special Version for 哥哥

- 这次界面要用中文写啦。

Question

- 才发现mac下的idle是不能写中文的...

Solution

> 1. IDLE 下中文输入法失效。这是因为 IDLE 界面使用的 Tkinter 需要依赖 Tcl/Tk，而系统自带的 Tcl/Tk 版本太低，造成了不兼容的问题。
>   可以下载新版本
>   或者用homebrew安装 `brew install tcl-tk`
>   via mac下的 idle为何不能输入中文?该如何解决?

测试后，还是不行

> 1. 根据IDLE的WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
>   Visit http://www.python.org/download/mac/tcltk/ for current information.
>   If you are using OS X 10.9 or later and a Python from a python.org 64-bit/32-bit installer, application windows may not update properly due to a Tk problem. Install the latest ActiveTcl 8.5.18.0 if possible.

发现是要下载8.5版本，而不是更新的8.6版本才可以。搞定。

Question

- 程序有中文需要declare

Solution

添加 `# -*- coding: utf-8 -*-`

Question

- 想要能保存多条日记

Solution

用while loop和break

代码

```
# -*- coding: utf-8 -*-
import time



                                                # ----问候----
print('哥哥，我是你的日记。')
print('你今天有什么想和我分享的吗？')
print('你也可以按"回车键"来阅读我。')

diary = raw_input('>>')



                                                # ----输入回车键，阅读日记----
if diary == '':
    diaryFile = open('diary.txt')
    diary = diaryFile.read()
    print('===========日记============')
    print(diary)



                                                # ----输入文字，开始写日记啦 ----
else:
    diaryFile = open('diary.txt','a')

    write='y'
    while write=='y':
        diaryFile.write('\n' + time.strftime('%Y/%m/%d')+ ' ' +diary)

                                                # ----还想继续写日记----
        print('已经帮你记下来啦。你还有想和我说的话吗？(y/n)')
        write=raw_input()

                                                # ----不想写啦----

        if write !='y':
            break

        diary = raw_input('>>')

                                                #----说再见----
    print('谢谢你和我分享这些。')
    print('再见。我会想你的。')

diaryFile.close()
```

Question

- 可不可以不通过命令行，而是双击直接运行呢

开心的反馈

把这个小小程序给哥哥用，哥哥不停的笑，写下了一行日记

> 2015/10/21 今天很开心，亲爱的老婆为我设计了个程序

喵，我也觉得很开心呢。心里超甜蜜！
顿时觉得一起分享才是更开心。
当他更明白我学编程是在做什么，也感受到我做的时候也会想到他，这样的感觉其实更好呀。

喵，不过哥哥平时也不写日记滴。
偶问了哥哥，他最想要打游戏可以自动练级的程序。
我现在倒是没有这水平啦，不过我觉得这种可以automate的是事情倒是很值得交给programming去干。
喵，继续学习，继续创造一些更有爱的东西啦。

Week4:意外与改变

Surprise与不插电试验

> 更新日志
> 20151025 创建

喵，这周最大的surprise当属发现自己怀孕了。
这种感觉真是难以言喻。
兴奋，开心，又有些紧张，纠结。
还有，我突然又开始想，怎样度过这一生，会觉得不那么遗憾？
面对未来，有一些害怕，又有一些憧憬，就是这样复杂的感觉。

这不得不让我想到两个问题：

1. 我希望给baby一个什么样的生活？什么样的生活态度？
2. 这三个月的编程我还学得下去么？

关于生活与生活态度

第一个问题，我觉得这个问题其实是问我自己，我想要为自己创造一个什么样的生活，什么样的生活态度？

1. 不去怪别人，我已经是大人了，我要学着为自己的人生负责。
   家庭环境，校园环境，社会环境，好像一路走下来，把问题归结于外界是很容易的一件事。但是，除了发泄一下情绪之外，并不能带来实质上的改变。
   我好像总还是以为自己是小孩子，是对很多事情无能为力的小孩子。而且虽然受困于这种无能为力，却还是不想长大的小孩子。
   因为，长大，在我潜意识里，还是一件充满痛苦和辛苦的事情，在我没有真实的感受到长大的自由和快乐的时候，总觉得还是害怕。
   而真的要进行角色转变，要为人父母的时候，突然就多了一份责任，也因这份责任多了一份勇气。我发现自己其实已经是大人了，也在学着尝试着用更成熟的方式去面对问题。
   当生活不是你想要的，怎么办？
   去创造，去尝试，去改变。去想尽一切办法。
   我觉得，其实我是有能力去解决和应对这个问题的，虽然还不清楚具体的方法，但是，探索，就会有可能。

2. 身教>言传
   其实我也不清楚baby会怎么想，不过，我一直觉得身教的力量>言传。
   我是想传递一种无力的观念，还是想传递一种创造与好玩的观念？
   想要baby好好生活，不是去演什么“我一切都是为了你好”的苦情戏，我为你放弃了什么什

么，最好的方式，是自己好好去生活。
虽然我不知道baby会怎么想，但是我了解我自己，从创造一个自己喜欢的环境开始，从创造一个自己喜欢的生活方式开始。
想一想，我是一个baby的时候，我希望我有一个什么样的成长环境。从这里开始。

3. 满足了自己的期待，才不会把期待放在别人身上
这也是我学钢琴的时候意识到的。那时候总觉得自己有了小孩会想让小孩学钢琴，喵，其实这是想完成自己未完成的愿望。
当自己开始学琴的时候，自己做了自己想做的事情的时候，就会觉得我以后的小孩想学什么学什么，反正钢琴的执念我可以自己解决了。
想一想，我对未来baby的各种期待，是不是也是因为我没有做到，所以把自己未完成的期待放在了小孩子身上。
但是，既然连我这种有执念的人活了二十多年都还没做到，为什么要苛求一个小孩子去完成自己的愿望呢？
其实，看到自己对baby的期待，也是映射出了，自己对自己的期待，为什么不自己学着去做呢？
满足了自己的期待，才会学着放下，也会更懂得宽容。

What to do？

让自己学着热爱生活，学着满足自己的期待，学着创造一个自己喜欢的环境。
why？因为自己自己热爱生活，才能传递给baby真正的发自内心的积极态度。
因为自己满足了自己的期待，才不会把自己的执着放在baby身上，自己未完成的想法，让自己去做。
可以把这个看成一个新的restart。
学着用自己的能力，用自己的大脑，用自己的资源，创造一个新的solution.

回到编程的问题

这个问题看起来是不可调和的。
喵，就像哥哥说的，都怀孕了，上班本来就对着电脑，现在下班还对着电脑。
想编程，也不急于这一时，过了这一年，以后再学也可以呀。
喵，可是自己却又觉得，这一次好不容易学得有点感觉了，一切正变得更有意思起来，本来真的是不想放弃的，可是，我真的是要放弃吗？即使我不想放弃，当上机时间锐减的时候，我还能跟上进度么？我很怀疑。
可是，我又觉得哥哥说得很对，自己也觉得老是对着电脑不好，人还是要面临一些权衡取舍的，不可能什么都要，现阶段更重要的顺利的让baby生下来。

不过，再换一个思路来看：
其实哥哥是想让我有一个更健康的生活方式。
更健康的生活方式，和编程是不可调和的吗？好像也不是。
其实自己也觉得大晚上的玩电脑确实有种上床了还睡不着的感觉，玩久了身体不舒服，眼睛也不舒服。这些也是本来就存在的情况，只是被我选择性的无视了。
这不也正好是一个机会，让我学着改变吗？
我想了很多办法，比如用看书的方式，比如打印，比如用口述的方法写文章，比如在纸上写代码。想一想N年前，上机特别不容易，不也有很多人在这种情况下学会了编程么？

我学编程是学什么？编程的思维方式。
那么编程的思维方式是什么？解决问题，不求完美，而是抓住主要矛盾，用最小代价解决问题。 对我来说，编程的思维方式，就是让我看到有另一种解决问题的可能，学会创造性的去解决。

既然这样，不如让我试一试啦，就当是好玩啦。

1. 本阶段主要任务：健康的生活方式
2. 学会运用编程思维：解决问题 创造性 不求完美 不断修正

不插电的编程试验

其实也不是完全不插电啦，平常的生活也还是可以用电脑用互联网的。只是要学着减少使用时间而已。

编程思维（最重要）

学会用编程思维解决生活问题，不局限于编程，很多事情都可以拿来练手

- input
  - my resource
  - my time and energy
  - someone who could help me
- output
  - things I wanna do

写作练习（输出是更残酷的输入）

writing offline:

- 手写，拍照上来
- 口述
- 手写，不断重构，把重构了几回后的文再打上来

编程练习（实践也不可少）

编程练习offline：

- 看书
- 手写书后练习和代码
- 头脑练习读代码和思考

上机（减到最少）

- 做每周任务
- 写每周教程
- 与大家互动
- 自己想测试的手写代码

当时间和精力有限，当你遇到一个坑 - Week2课程笔记

Zoe，昨晚折腾了一晚上，折腾在各种pip install, brew install,import中，各种莫名其妙的Error中，是不是有种心力交瘁的感觉？

是呀，本来现在可以用电脑的时间就少了很多，本来准备晚上把代码写了，结果晚上几乎什么也没干，想装的东西也没装上，还不知道错在哪里，这感觉多纠结 >_<

好啦，正好新鲜出炉了Week2的视频，让我们一起来学习一下，顺便也看看你现在遇到的问题有什么新的解决思路啦。

最小代价解决问题，专注于解决主要矛盾

Zoe，还记得自己本来打算晚上干什么的么？

喵，偶想学习GUI来着。然后，踩到了一个坑，然后又踩了一个坑，然后又踩了一个坑...然后，一晚上就就在坑里了。

想学GUI的话，这些坑是一定要踩的么，还是可以绕过去的？

那倒也没有，其实就是觉得PyQt5看起来很好用的样子，我觉得装起来也不会那么复杂。
而且，就算有些问题，我以为google一下应该就解决了，谁知道这么复杂，看到的是个小问题，扯出来的却是一条大象T_T

好吧，我们踩坑的时候，其实都不会提前知道那是一个坑。在越陷越深前，学会止损也是一门艺术。

喵，是的，我记得大妈有说到过的。
不过看到问题就一脑袋钻下去了，完全不记得这是不是我本来要解决的问题了。
然后浪费了好多时间后，回过头来才发现，好像捣鼓的那些东西完全偏离了主线了。

好啦，吸取惨痛的经验，让我们重温一下——如何止损？

> **30**分钟搞不定的坑，填这个坑真的有必要吗？还是可有可无？是否可以换一种方式，学会是否可以求助别人，是否可以绕过这个问题，是否可以过一阵再再来解决？

下次看到坑的时候，不要那么恋恋不舍啦，Zoe。

明白，以前觉得解决一个坑是学习，是成长，总觉得想要搞定。
搞不定也钻牛角尖，不想出来，就想能搞定。
不过现在想想，搞定了又怎么样呢，完全还没有触及到问题的主要矛盾呢。
有坑不要乱跳，因为有时候爬出来还是需要很多时间的，甚至是远远超过自己想象的时间。
其实这也是经济学里提到的机会成本，这些时间本来可以用来干些更有意义或者更有趣的事情的。
当时间和精力有限的时候，要学会专注在问题的主要矛盾。

磨刀不误砍柴工，从改变自己的基本思维习惯开始

Zoe，你觉得大妈的解决问题的方式，和你自己有什么不同？

喵，大妈好像用官方文档就可以解决问题，我是比较怕这种的，觉得读起来好无聊，而且一大堆英文夹杂术语，完全不知道在说什么，看着打瞌睡。
好像自己比较习惯于网上搜课程或者别人的经验，让自己可以step by step的去做，因为自己比较读得进去一点。
包括对命令行的使用，我之前也是比较怕怕的，也想不到那么多神奇的用法。
google的搜索，其实也没有好好去琢磨过，还是凭感觉在用。
但是确实有种感觉，真的非常不一样。

那你觉得大妈的这种解决方式，有它的优点吗？

嗯，主要是搜索别人的经验教训有时候并不准确，或者内容过时了，自己对照着做，有时候踩了坑都不知道。但是官方文档里，其实很多都还是有提到的，而且很准确，也比较全面。
另外今天翻python的官方文档，发现真的提供了很多学习资源，之前都是自己从google上到处搜集，才发现原来这里就有一个值得学习的知识库，之前根本就没发掘到这么个地方。

你觉得自己之前没想过读官方文档的原因是什么呢？

主要还是界面看上去很无聊，读起来比较枯燥啦。现在网上音频视频图片这么多，看着这么一个不够漂亮的界面，我有点走神。
不过更重要的还是自己没有体验到过用官方文档的好，虽然别人都这么说，但是没有具体的感受过这种好。

就是说，对你来说，用更感性的体验式的方式，更能够影响到你，让你觉得想去改变？

嗯，对。当没有体验到之前，我不知道它的存在，也不知道它的好，它对我来说只有一种抽象的感觉，只有一种我明白这个道理，但是提不起兴致去做。 但是如果让我看到，让我体验到，感受到了它的不同，我觉得这才是促使我想去改变的真正原因。

觉得思维方式的改变，对解决问题有什么影响？

可以从另外一个维度去思考问题，解决问题了吧。
比如之前完全没有想到一种可能性，觉得问题无解，但是换一个人换一个视角，感觉就很不一样了。
还有就是可以更好地举一反三了，不是生搬硬套答案，而是知道怎么去思考和解决这类问题了。
因为其实也是看到了自己的思维盲点，原来不知道自己不知道，现在可以发现原来自己看不到的一些东西了，然后想问题解决问题的角度就完全不同了。

那你觉得，什么方式比较对你的口味，能让你去改变自己以前的思维方式？

我觉得最重要的一点，还是从看到开始。就是要从看到自己不知道开始。不然完全都不知道哪里有问题，就更不知道如何解决了。其实也是我上面说到的体验啦。

1. 体验到有别的可能性。这个还是需要多和不同的人去交流，多读不同的书，有更多不同的思想碰撞，这一点真的很重要，能让我看到，原来还可以这样去做，去想。
2. 体验到这种方式真的用起来很棒。就是要自己能够体验到，这种新的方式带来的愉悦。自己要去刻意练习，要让自己在这个过程中开心，真正感受到新思维方式带来的快乐。当然这有个循序渐进的过程，不过可以慢慢往新的方向去尝试。

有同伴，真的会让人更想要继续下去

发现你这次其实没有之前那么想要半途放弃了呢，哪怕是你现在很有放弃的理由，**hoho**，但你还是想再试一下，你觉得是什么原因？

第一个，不想跟不上进度啊，看到大家那么火热朝天，顿时觉得自己也该好好努力啊。
第二个，有大家一起交流，顿时觉得很温暖，觉得不是一个人在战斗啊。
第三个，有什么想法，可以去和别人分享，而不是自己默默地想，觉得自己做的事情更有意义啦，写东西抖更来劲啦。

之前有过类似的体验么？

嗯，我觉得最接近的就是在ToastMaster里的时候，那时候看到一起进来的Joe，Wind都那么棒棒的成长，顿时觉得自己也要好好努力啊。
而且觉得一起进来的，看着特别亲切，可以一起聊，去蹭饭，感觉多了个朋友的感觉，很放松，很温暖，那段时间和开心。现在想想，也还是蛮想念那时候的，自己的成长也蛮快的。

既然体验这么好，为什么没有继续下去呢？

一个也是因为大家毕业了，熟悉的人都不在了，感觉没有之前那么亲切。加上不在一个城市，跑过去还是比较累的。
还有一个太久没练习，看到后来的人都进步那么大，自己都有些不好意思去了。就是有种自己不如别人的压力吧。这个我觉得是最主要的。
另一个就是确实是需要一些时间精力去准备的，真正做起来还是比较累的，有时候觉得没有这么多精力去做。

其实也还是在一个比较开放友好的环境呀，为什么不如别人会对你有这么大的压力，让你想要逃避，而不是一种动力呢？

其实同期的人，还是比较能给我一种动力的。
但是看到别人后来居上，还甩了我N条街的，有种继续不下去的感觉。
因为这让我真实看到了自己的差距和不努力啊，感觉好羞愧。

后来居上，被甩**N**条街这种事情，其实很多地方都会发生吧，有没有想过怎么更好的去面对这件事情？

嗯，自己也发现，这也是我的心结之一。
现在想到了几点：

1. 记录自己的成长，让自己看到，其实自己也有在每天一点点进步的啦。而不是老觉得自己这不好那不好。
2. 看清自己真正想要的目标，专注于自己最想要的生活，而不是别人想要的生活。把目光从和别人比，收回到靠近自己想去的方向。
3. 学习他人是怎么做到的。如果这也是我想做的事情，有没有什么启发？有什么我可以去学习的，让自己也有这样的能力？
4. 最重要的是，让自己的生活真的往自己喜欢的方向迈进。真的觉得自己爽到了，真正触动自己内心了，真的觉得自己满足了，其实也就不会那么和别人比来比去了。

之前做事的时候，有没有想过借助这种同伴的力量，社群的力量？

没有，比较喜欢一个人做。比较不喜欢和太多人打交道。

这次体验有什么触动没？

想要为自己创造一个新的圈子了。
因为圈子真的会带着你跑，往你喜欢的，或者不喜欢的方向跑，真的这种力量很强大。
我觉得现在之前上班接触的人和思维，会让自己变得比较狭隘和无聊，比较看不见新的可能

性，因为他们本身的生活方式就是我想逃离的，每天被这种思维熏陶下来，只熏陶除了无奈地感觉，而不是充满生命力的感觉。

具体一点，想怎么去做呢？

从易到难。
先从改变自己阅读信息源开始。从阅读开始就要拓宽视野。
再改变自己的互联网接触圈，现在其实就是一种尝试。
再从自己的兴趣爱好出发，拓展线下的圈子，反正是接触一些自己喜欢的感兴趣的人，应该不至于那么头疼的啦。
总之，学着为自己创造一个新的喜欢的环境来。

这一周，如何行动？

1. 问自己，如果今天只能做一件事，我想做的是什么？ 让自己focus到最重要的事情上来
2. 问自己，这种方式和我之前有什么不同？我之前没有想到的是什么？我之前遇到困难是因为没有意识到什么？学会看到自己的盲点
3. 问自己，我喜欢一个什么样的环境？我喜欢什么样的生活方式？学会为自己创造这样的环境

小小日记-桌面版

思考一

完全不懂GUI，所以需要一个教程来入门一下。

- 本来准备用Udemy的一个关于PyQt的课程的，但是发现完全卡在了第一步安装上了。
- 改用python自带的Tkinter，先绕过安装的这个坑，开始练习再说。
- 搜索tkinter tuturial，选用了一个图文并茂并且看起来比较短的教程先学。
  > 参考资料：Python Tkinter
- 发现时间不够用，所以先挑一些这次项目会用的相关章节先看。

思考二

关于GUI

我希望我的GUI，有可爱的图片，字体好看一点啦。
既然是GUI，就发挥一一下GUI图形界面里图形的作用。
其实我觉得GUI就是要美美美，就是暂时实力还比较小白。

关于网页交互

对网页交互这一块比较感兴趣，因为觉得网页端的跨平台性比较强，手机和电脑可以通用，而且也不用安装python就可以运行。
这一部分有待发掘。

思考三

打开程序后，显示两个Button（带有图片显示）

- Write
- Read

Write版面

- 输入框
- 三个Button
  - Save

  - Read
  - Leave

Read版面

- 阅读框
- 一个Button
  - Write
  -

尝试一

添加图片界面

- 搜索read和write的图片，改成gif格式，宽度改成200
- 想使用带图片的button，google搜索，貌似和self.photo命令有关
- 搜到一段还有点类似的代码，button是在图片下，我觉得也可以

  > 参考资料：[Tutor] Images + Tkinter

- 尝试着改一改，有个页面雏形
- Button改成了中文，程序开始要添加一行 `# -*- coding: utf-8 -*-`

添加界面跳转

- 想象中有几个界面，想在几个界面中进行切换，开始是搜索的tkinter button change window，后来发现其实我的意思是在几个frame中切换
- 在stackoverflow上有段超棒的解答，虽然我不太看得懂，但是看到程序运行出来的效果，我就觉得写得真的好棒

  > 参考资料：Switch between two frames in tkinter

- 尝试把上面完成的图片界面程序和这个界面跳转程序连接到一起

添加读日记界面

- 在尝试在读日记界面添加text的时候，遇到global name 'END' is not defined，搜索后发现END要改成tk.END就可以了

  > 参考资料：tkinter NameError: global name 'END' is not defined

- 整合上一周的的阅读diary的代码

添加写日记界面

- 添加input界面-Entry Widgets,还是有问题，尝试搜索self entry tkinter,发现了上一次回答界面跳转问题的人对方面也有回答,而且自带了get button，也是我等下会用到的
  > 参考资料：Tkinter Entry “get” function is returning nothing
- 发现Bryan Oakley关于tkinter的回答都很棒，值得细读
- 修改了一下Entry的padding，看起来更舒服一点 `self.entry.pack(ipadx=100,ipady=10)`
- 保存完自动清空内容 `self.entry.delete(0,tk.END)`
- 整合上一周的的书写diary的代码

继续修改

- 发现阅读diary的内容没有正确展示出来，后来发现是因为height不够，后面的内容没有显示
- 考虑增加scrollbar
- 根据前面的tutorial中的scrollbar部分，一行行的尝试添加代码，看哪里可能出错，修正
- 添加 `from Tkinter import *` 减少报错
- 考虑修改阅读版面的字体

中文支持

- 发现不能保存中文，将get()语句修改 `diary=self.entry.get().encode('utf-8')`
- 发现可以保存复制的中文，但是中文不能打字
- 发现在python的idle中运行时可以打中文的，但是sublime中不行，推测是调用的tk版本不同的问题
- 但是在idle中运行程序，点击左上角关闭感觉有些死机，sublime中没有这个问题

问题

- 发现阅读diary中内容不是实时更新，而是启动时更新的,觉得这应该和init有些关系，待解决
- sumblime直接调用不能打中文
- idle调用不能直接关闭程序（貌似死机）

Week5:节奏

小小日记-Net101

尝试一

- 完全不懂本周说的到底是什么，看到芝麻星上有提到socket和UDP，觉得应该从这里入手
- google搜索python socket tutorial 发现一个简短又看着比较舒服的教程 Python socket network programming,对server和client有一个大概的感受
- 文末有推荐另一个教程Python socket – network programming tutorial
- 看着看着才发现上面说的都是tcp，关于udp的教程在这里Programming udp sockets in python
- 搜索python tcp udacity，看在udacity上有没有合适的资源，然后搜到了Computer Networking中的 What the Course is NOT About，意思是说学这门课需要有一些基础知识，下面给出了一些学习链接，我觉得可以从这里入手
- 瞬间有了一堆学习材料，其中最后部分才是和python相关的，而且最后两个链接也是上面搜到的binarytides网站的，所以决定先从这里入手
- 还是觉得官方文档入门不够友好，适合懂得一些之后去查阅，但是给没有一点概念的人看太费劲了

Instructor Notes from Udacity

Some suggestions from Piazza:

- Network Programming Study Guide
- TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994, ISBN 0-201-63346-9
- Stanford Course: An Introduction to Computer Networks (Fall 2012)
- Computer Networks: 5th Ed. by Tanenbaum & Wetherall

A quick google search turns up these slides for TCP/IP basics:

http://www.slideshare.net/sanjoysanyal/tcpip-basics

And a few python tutorials on socket programming:

- http://docs.python.org/2/howto/sockets.html
- http://www.tutorialspoint.com/python/python_networking.htm
- TCP: http://www.binarytides.com/python-socket-programming-tutorial
- UDP: http://www.binarytides.com/programming-udp-sockets-in-python

尝试二

- 使用udp要用到netcat,而不是telnet
- 安装netcat `brew install netcat`
- mac里用netcat要用netcat指令，而不是nc和ncat，如 `netcat localhost 5000 -u -v`
- 通过上面的UDP学习文档，了解到了一个简单的socket和client的效果

尝试三

- 尝试与之前写的小小日记CLI版本结合起来
- client写一条message server能够回复print出时间+message
- 把时间改成了python2版本的了

```
import datetime
today=datetime.now()
print today.strftime("%y/%m/%d")
```

- 中文测试也是ok的
- 尝试在回复的同时，在server端打开diary.txt，并记录日记
- 发现改了代码还是无法在目录下建立diary.txt,后来发现while loop中，它是记录到我的Zoe文件夹（也就是默认的home路径下面去了）
- 推测，因为我的运行方式是 `python /Users/Zoe/GitHub/socket/diary-server.py` 尝试切换了目录再运行
- 顺利整合了原有功能
- 尝试多个客户端登陆，没问题
- 添加了代码，可以在diary.txt中体现是哪个端口登陆的 `diaryFile.write('\n'+today.strftime("%y/%m/%d")+' client['+str(addr[1])+'] '+diary)`

感想

- 喵，看到大家交作业好神速，觉得不可思议，因为当时还处于一头雾水什么也不知道的情况。结果自己一尝试，好像也没有想象中难耶。
- 吸取上次教训，上次开始就想做一个有点复杂的功能，给自己挖了一堆坑，完全出不来啊，写的代码也不知道是个什么，这次就是从实现最简单的功能开始，过程更流畅。
- 吸取上次教训，周三开始写代码，交作业急急忙忙，弄到很晚，还写得很烂。这次也是受到大家的影响，发现早点动手最最好，周六开发出来第一版，整周的节奏才更加的不紧不慢 ^_^
- python编程就是这么简单，一点点代码就可以搞定问题，噢耶！
- 回到熟悉的命令行模式，再次感受到命令行开发真的比GUI简单好用。

节奏与不断靠近-Week3课程笔记

写了一个多小时，却在要发布的最后一刻，发现记录莫名其妙的消失了，这真是一种叫做没有备份的痛！ 重来...实在没劲了，这次写简练一点...

前言

从上周遇到的坑说起。上周周三临时抱佛脚的交作业的体验，不想再重来。回想起来，发现自己有两个主要问题：

- 1.周三晚才开始着手写代码，时间匆忙，弄到很晚，自己也觉得很累。其实是一周的时间节奏不对。
- 2.一开始想着去实现一个对自己目前能力来说有点复杂的功能，结果自己的思路不太清晰，还踩了一些自己也不太懂的坑，最后写的程序功能虽然实现了，其实自己有点没看明白。

节奏

早在Python星际旅途开始之时，舰队的首席布道福音师——大妈，就给了每个人一段关于节奏的提示。

- wd0 进行/开发/实现 任务
- wd1 整理私人教程, 四下串门, 交流经验
- wd2 测试/完善/回顾 任务
- wd3 小息, 总结前周
- wd4 公开课, QA
- wd5 新任务接手, 背景调查, 先期检验性开发
- wd6 理解/分解/尝试 任务->定好方案

不过，虽然Zoe看到了这段话N遍，但是想到自己平常经常有加班，遵照这个节奏好像也不是那么适合，所以还是随心所欲的按自己的节奏来。
直到上周任务纠结的交付后，这才痛定思痛，每周可不能这样过呀。
Zoe重新思考了一下每周的任务：
主线任务（每周编程项目）

- 1.背景包-学习背景知识和教程
- 2.行动包-尝试第一轮编程
- 3.拓展包-到处转悠，看有没有新的其他，尝试修改第二版

副线任务（每周笔记）

- 1.背景包-看视频教程
- 2.行动包-写笔记

考虑到周一周三可能会被加班，Zoe发现，起码周日就得写出自己的第一版程序了，可是实际当中她周末还停留在看上周四视频和写笔记的副线任务中，主线任务几乎没有动静，难怪会觉得时间很赶了。
Zoe重新翻到了自己在开学典礼上的记录的笔记：

- 错误0：花了太多时间学习在那些不是特别需要的东西上
  现阶段有些书是不必要看
- 错误1：没有立即开始写代码
  最开始就浸泡在问题域中，用自己的思想和方法改进自己的错误，才能获得自己的经验
  喵，这周，Zoe决定改变方式，先完成主线任务再说。正好这周的任务不难，周六顺利完成了第一版程序。喵，这周不用抱佛脚了，万岁。顿时觉得这周轻松了，也可以静下来进行一些别的探索和思考了。

  > 大妈说： 是也乎,(￣▽￣) 这也是俺为毛从 0w 就开始反复强调节奏的原因哪! 只有每个节点儿都在节奏上了, 才有信心进行不断的优化,反思,教程积累, 从而获得自个儿的经验以及目测力 ;-)

这一次是真的亲身体验到了不同节奏的利弊。
真的觉得，自身的体验是一个很重要的过程。不然，即使别人告诉了你很好的方法，你也不见得懂得它的好。好东西放在面前会视而不见，是因为没有体验。

面对未知，不断靠近

在Python之旅开始的时候，Zoe写下过一个疑问，解决问题有两种思路，一种是直接从最终结果来反推中间过程的方式，另一种是从最简单版本不断靠近到更高级版本的方式，看起来各有利弊，到底哪种方式更好呢。
当Python之旅进行了一个多月的时候，Zoe有了自己的答案。 当问题是比较简单的、自己比较熟悉的领域的时候，两种方法看起来是差不多的。
但是当问题的是自己未知的需要探索的领域，是一个很复杂的领域的时候，后一种方式的优势就更明显了。
这是因为，未知领域，有很多不可预料的因素。有很多环节，是现在的自己无法看到的，无法反推到的。有很多的坑，是自己不知不觉就会踩进去的。有太多东西，是在自己的意料之外，计划之外，掌控之外的。这可能让整个系统的逻辑混乱，无法正确的运行，连基本的功能都难以实现。
而后一种方式，前进的每一步相对起来比较简单，因为每次只需要踏出一小步就可以，难度不是那么大。所以即使是探索未知领域，但是因为只需要小小一步，所以容易取得突破。同时，每一步前进也有一个坚实的基础，因为之前的每个版本都实现了一些基本的功能。

启示

举一个困扰我已久的关于工作方面的例子。 上班上得很纠结，想离职，又不知道能到哪里去。

想一想自己的节奏：
上班的时候 -想快点度过这一天，快点下班。
下班的时候 -终于可以做点自己的事情了，想，我要做些什么呢？+我要放松压抑的心情
夜深人静的时候 -我不想过这样的生活。
喵，这个节奏是不是有点不太对头。虽然情绪很强烈，但是请问我的主线任务在哪里？看起来，是木有-_-||| 专注纠结三百天。
就好像，我要编程了，突然觉得我要先看一堆书，打好基础（其实是一堆还没有必要看的书）。N天过去了，请问你开始编程了么？嗯...我觉得我还没有准备好...
就好像，我想离职了，突然我觉得我有很多要思考（其实是一堆现在还没有必要的思考）。N天过去了，请问你为离职做了什么呢？嗯...我觉得我还没有准备好...

因为，我真的不知道要做什么，我不知道未来是什么样的，我也不知道要怎么从未来反推现在的要做的事情。
OK，那说到刚刚学到的新的领悟——面对未知，不断靠近。

虽然不知道要怎么走，但是，
可不可以尝试学一门现领域之外的课程，探索一下其他的方向？嗯，尝试了加入编程课程。
可不可以给自己一些新的信息源，了解一下领域之外的新的认知？嗯，在学编程的过程中，确实会接触到新的信息源，看到新的成长故事，看到新的交流方式和协作方式。
可不可以尝试着将学到的新东西，运用的自己的生活中来？嗯，每周写日记，就是在反思这么个事情来，也就是编程思维对生活的应用。
可不可以尝试着认识一些新的领域的朋友？嗯，这个也不难，这么多棒棒的老师和同学们。
可不可以尝试着做一些自己以前没做过的事情？嗯，gitbook写教程，感觉挺好。
……

当面对未知，直接冲上去，可能会不知所措，可能会撞墙，也可能会掉坑。
但是，从每一个小小的“可不可以”开始，踏出一小步，再一小步。
当目光聚集在自己能做到的事情上，而不是不能做到的事情上，就能够更坚实的迈出向前的一步。
这就是编程思维教给我的面对问题的思路。

Week6:Web世界

小小日记-Web101

尝试一

- 读了bottle的tutorial，感受到几行代码能写出一个web的速度感，很震撼。
  不过读文档还是有些看不懂，主要是感觉和自己具体应用层面又还有些差距。
- 继续搜tutorial，发现了官方的Tutorial: Todo-List Application,觉得这是个很好的尝试，其实todolist和diary有些相似，都是需要一些添加删除阅读之类的功能嘛，所以从这里开始啦。

尝试二

首页

用html语言就可以搞定了

- 问候语
- 写日记和读日记的link

读日记

希望能够增加back home 的link以及\n能够自动换行

- back home的link直接用html添加即可
- 关于\n变成 <br \> 自动换行
  尝试了N次之后，发现用template的时候，<br \> 是直接显示出来了，而不是作为html语言起到作用，
  所以直接用字符串的链接，没有用template来显示日记了

  ```
  output='<p>====日记====</p>'+diaryContent+"<br /><br /><a href='/writing'>写日记<a><br /><a href='/'>Back Home<a>"
  return output
  ```

  觉得这里还有待改进，不过总算实现了自己想要的效果

写日记

直接用了todolist的tutorial中的new-task模块， 稍微修改了一下名字之类的。
调用这部分内容，其实就是把上一节的net101中的 `newDiary=data.strip()` 改成了 `newDiary=request.GET.get('newdiary', '').strip()`

尝试三

- 做完之后发现我完全无视了另外一个要求， `同时兼容 3w 的 Net 版本的命令行界面进行交互` ,囧囧囧,感觉自己完全跑题了
- 把上次的做的diary-server,diary-client拷贝到这次的文件夹下，发现其实也很顺利，照样能用，反正都是调用的diary.txt的数据嘛
- 虽然跑题了，不过好歹两套系统都是独立可用的...

后记

- bottle写网页居然这么简单，几十行代码就搞定了，写完有种难以置信的感觉
- 感觉网页还是不好看，可是bootstrap暂时也不知道怎么用上去的好，所以先试验性的在greeting页面中加了一点html和css的内容。
- 真的想把bootstrap能用上去，但是搜到的好像是flask和bootstrp的结合比较多。
- 真正要架设在网站上，还是有要改的地方，如果不是调用自己本地的diary.txt,而是改变网站中的diary.txt，该怎么改呢？这回肯定不是用import os来解决了。有待思考。
- 通过bottle，不同的用户可以方便的拥有属于自己的界面，但是以后放在网站上要使用的话，起码要添加密码认证，保护自己的内容。

给你更大的世界 - Week4课程笔记

喵，每天都像在晕车一样，这就是传说中的孕吐。
感觉自己很多时候已经不太能用脑子思考了。每天基本都是加班回家就睡觉的节奏。
还好上周有先见之明，周末就把任务做了，完全放了大半周的假也还跟上进度了。
现在也不给自己定复杂任务啦，赶上进度就ok。每过一周对自己来说都是小小胜利。

这段经历也让我体会到身体的重要性了，有健康的身体才有能好好思考的大脑啊。
我都想到下一阶段想主攻的就是锻炼身体了。

终于又到了相对清闲一点的周末，看到这周视频的时候还是有些兴奋。
编程思维很好玩，能学着运用到生活中会更好玩。

以有涯随无涯，要学会取舍

互联网太丰富，太容易迷失在这片互联网的海洋中，
从一个链接跳到一个链接，想挖的坑会越来越多。
所以，我们要学着用最小代价去解决问题。

学编程的这一个多月，已经让我感受到编程宇宙的辽阔，这还只是人类世界中其中一个小小领域而已，更不要说更多的领域，到底有多么的深不见底。
这一个多月，让我更深深的感受到了自己的无知。感受到知之有涯，不知的无涯。

更大的世界，为了不至于让自己迷失的更彻底，我们该怎么办？

1.回归源头。万变不离其宗。

正如大妈说curl的时候，说到的，无数软件是由有限的命令行组合而成的。
向源头进发。我们就能从纷繁复杂中看到更简单的本质。
正如我们接触命令行的时候，看到操作系统在后台运行的另一个世界。
包括这几周过来，有一些话会反复的出现，每周读到都会有新的意思。
有时候，会觉得很多东西都未曾改变。

2.学会取舍，路不止一条。

钻牛角尖的一定要搞定这个问题，是真的对解决问题有帮助吗？有别的可能吗？
我觉得对自己思维的改变，就是感受到路不止一条。
我这么执着于钻牛角尖的想解开这个问题，是因为我觉得路只有这一条。我觉得解不开，我就无法继续走下去了。

真的是这样吗？一路走过来，感觉在编程的过程中，遇到了很多坑，也弃掉了很多坑，但是居然也这么走下来了。
如果每个坑我都细细琢磨，我估计我还掉在某个坑里没有起来呢。
条条大路通罗马。现在的自己能慢慢感觉到，也许事情还有别的解决方式。这个方式不行，就换一种。
不要执着于方法，而是想想自己要做什么，希望有什么样的效果。

每周10小时，为什么可以有这样的改变？

之前其实是想，每周10小时，原来可以有这么大的改变。
我觉得我都可以每三个月给自己开展一个有趣的计划了，学着用三个月，每周10小时的时间去认真去做一件事情。我现在觉得3个月的时间其实是完全有可能学到一个新领域中的很多东西的。

但是，我现在在想得是，其实并不仅仅是告诉自己，每周抽出10小时的时间来，
而是为什么在这里，可以真的做到改变？为什么之前的尝试并不成功呢？究竟有什么不一样的地方？

1. 任务设计——原来我可以做到

我觉得每周做作业之前，都觉得自己是要在本周完成mission impossible,头都要大了，但是真正去做的时候，发现其实又没有那么难。

但是如果自己给自己设计任务的话，我觉得我肯定都是从一种比较平滑的方式来给自己设计的。不会给自己布置这么些让自己觉得搞不定的任务。
而且我觉得一般的小白的书里，给出的都是一些比较小白和简单的例子，不会扯到这么多领域，因为已经超出小白的范畴了。
但是我觉得在这次的学习过程中，我会发现原来小白也有能力做些相对不那么基础和简单的事情。 原来自己并不用限制自己的能力，这样的事情我也可以做到。

想到那时候khan学院，也是要利用有限的资源，给不发达的地区带去好的教育解决方案。看似不可能，但是换一种思路，会发现没有那么难以解决。

我觉得很重要的一点是，发现，噢，原来我是可以做到的。
其实每周的任务，涉及的内容看似很多，其实要实现的东西还是比较简单的，就是很基础的功能。 从我不行，到我可以，就像大妈再三鼓励我们的，其实这些能力我们一直都具备了，学会用自己的原力。

以前面对问题，会觉得太难了，小白的我做不到。
而现在，会觉得其实也不一定，说不定有别的新的办法呢。不会那么轻易的去否定一个可能性了。

任务难度的设计，其实可以不用像我之前那样的保守。 这种不舒服的感觉，正是因为自己接触了自己不熟悉的领域，也正是在这种不熟悉的感觉中，才有更大的成长性。

2.资源配合——视野与指引

我觉得少不了的，也是更高层面的指引。
不论是每周的教材，或者看大家的交流，还有芝麻星卡片给出的提示，能让我跳出我自己的视角，看到一个自己之前完全想不到的维度。
其实之前学udacity的课程也有这样的感觉，真的就是有一个懂的人，带你走一截的感觉好不一样，有一种带你俯瞰的感觉。他们接触到的资源，他们常用的工具，他们的思考习惯，对我们来说是非常有启发的。同时也更容易形成好的习惯，不论是思维上的还是行为上的。
所以真的要拓宽自己的视野。小小白也要学着养成专家的好习惯。

3.同伴与反馈--持续动力

自己对自己的承诺是容易放弃的，因为自己是比较容易受到情绪影响的，经常因为情绪低落，就停止了，就不想继续了。碰上一些杂七杂八的事情，心情不好，就一下子什么都不想干了。
然后再没什么人支持或者一起的话，有时候也会怀疑自己到底在干嘛。做的事情是不是没有意义。
我觉得能走到现在，收到的鼓励和反馈真的给我很多动力，每周看视频也有定时打鸡血的效果，主要是大家一块，在这个氛围下，自己没动力大家也会拽着你走，hoho。
被动进步的氛围，真的很神奇。

组合，联想，跨界，看到新的可能

这次大妈总结了一下python 能干的事情，让我又脑洞大开了一下。

- 1 批量。python几十行可以搞定，比如批量改文件名，发邮件，短信。
- 2 自动。所有能调用系统命令完成的事情，第三方服务，公开的数据源，API接口， 各种点鼠标干的事情，统统可以自动化。比如帮忙刷火车票、电影票 、双十一。
- 3 组合。自己组合你的数据和命令。比如想学摄影，电影每隔十秒，抽一帧照片，抽出主色调，压缩，制作色氛照片，看每个导演爱用什么颜色。 多种软件组合起来，极其特殊的需求自己搞定。

喵的，搞得我又有点兴奋了，我突然觉得自己手上有很厉害的武器，只是暂时还没明白具体该怎么用，但是真的有很多可能性就这么出现了。
学编程之前是不会想到有这些可能性的，只会机械的鼠标点点点，然后抱怨这系统怎么这么不人性化，为什么要做这么无聊的事情。
但是学了编程之后，就可以写好程序让计算机自己搞定这些事情，之前很复杂的事情，可能都会变得很简单了。

包括大妈提到的联想的例子，

- 0w，42行代码，整个软件日常80%的工作情况
- 1w，命令行，所有操作系统的后台，去到了背面
- 2W，看文档，一切软件的源文档都敢看了，去查了
- 3w，API，所有服务的接口你就可以了解到了。比如github的接口，几百个官方认证的服务ho

我觉得这联想能力一开，威力无敌啊。
这让我真正见识到了什么叫举一反三的能力。
全新宇宙等待你遨游。

现在觉得学编程太需要联想能力了，因为有联想能力，就会发现很多有趣的可能性。

Thinking：怎么让这漫长的九个月变得更有趣一些

把九个月分成3*3个月的小项目，做一些自己想做但是没有好好去做的事情。
记得从前说的左手音乐，右手计算机，还可以有写作当我一直的伙伴，感受互联网时代更自由的学习氛围，认识新的朋友。
有没有可能，让这些成为我的新的生活方式？
更进一步，不仅仅是线上的改变，更是线下新生活的改变，如何让这种改变渗透到线下来，而不仅仅是分隔明显的两个部分？
让这九个月，当成自己的一个reborn。

外一篇

Google Python Style Guide

1.看过吗
知道，但是没有认真细看过

2.看了几次
0.0001次

3.什么想法

- 英文读着费劲，发现有中文版
- 直接读还是读不下去，每次读一节好了
- 今天读了Indentation，之前好像都是用tab，这里提到用四个空格。
- 开始看缩进的例子一下子也没看出区别，后来发现是缩进的具体位置有所不同，真是训练眼力。
- 一边学了马上一边改自己的代码更好。

- 觉得自己读的方法还是太原始，读着没有兴致，期待发现更好玩一点的方式。

互联网的世界，让我们启程吧！- 搭建你的第一个python小小网页

受到liangpeili童鞋把网页搭建在互联网上的启发，
觉得既然都用bottle把网页都写出来了，也是时候把这个网页放在互联网上啦。
不用每次想用的时候，还得从自己的电脑上启动，或者从别人的电脑上还要下载python环境才能用。
直接打开网页访问，这才叫帅。
这也是我最开始想实现的效果。不然总觉得自己写的程序，给哥哥用，哥哥都不知道从哪里打开。
放上互联网，哥哥也好，妈妈也好，朋友也好，小白也好，秒懂。

发布平台的选择

开始是上传到自己的网站空间，后来想，不对啊，虽然我的python程序放上去了，又没人给我运行。
不像html放上去，就直接显示了。

然后联想到以前参加Rails Girls的活动的时候，用ruby的代码写的很简单的程序，也是放到heroku上就变成网页了。

猜想这个应该涉及到支持一个可以在后台可以运行python的平台，或者说后台经过某种部署后，python文件是可以以某种方式运行出来的。

当然，这个我也不太懂，看了官方的文档的Deployment部分。
了解了一下bottle的Deployment。
不过还是不是很懂，提到的一堆方式我都不熟悉，唯一一个稍微眼熟一点的，是Google App Engine。
推测这个用Google App Engine 是可以部署出来的。

获得一个粗略印象后，还是先用google搜索网站获取灵感。 搜索关键词‘bottle python web deploy’。
有提到Red Hat Openshift 和Simple Bottle Hosting: PythonAnywhere。

浏览了一下网站，在(PythonAnywhere)
[https://www.pythonanywhere.com/details/bottle_hosting]上提到‘From sign-up to a live Bottle app in 2 minutes...’，顿时觉得就这个了。
因为我确实看不懂一大堆的Deployment文档啊。

事实也证明，PythonAnywhere上搭建bottle网站确实很方便，什么都帮你设置好了。我要做的就是写一个Python文件，然后就等网站自动设置就OK。

在PythonAnywhere上建立自己的网站

1. 注册了免费账户，登陆之后，在右上角的Dashboard里，可以打开我们常用的几个面板。
    - Consoles
    - Files
    - Web
    - Schedule
    - Databases
2. 打开Web面板，选择add a new web app。
   填好后，会根据你选择的框架，如bottle或者flask， 自动给你配置好网页。
   现在你打开类似于username.pythonanywhere.com的网页，可以看到一个hello的网站已经生成好了。

3. 打开File面板，选择mysite文件夹，在这里可以看到具体的python代码。已经有了示范文件。
   只要将bottle_app.py改成你想写的代码就可以了。

4. 代码更新后，打开Web面板，选择那个大大的绿色的‘ Reload yourname.pythonanywhere.com’，你的新代码就生效啦。

我的网站和代码参考

网站 http://zoejane.pythonanywhere.com

代码 https://github.com/zoejane/python-anywhere/blob/master/diary-web.py

感想

- 放在互联网上，感觉比自己玩帅多了。发布出来的感觉真棒！
- 代码写得比较乱，主要还先把想要的效果搞定出来，写代码的好习惯后期还是要练习。
- 自己玩不如一起来分享。赶快加入互联网的世界吧 ^_^

Week7:那不存在的门

那不存在的门

真性情

今天，共事的一位姐姐要去别的支行了。
不知道怎么今天心血来潮的想起和她说一段道别，谢谢她教给我的，也想祝福她。
然后，写着写着，不知怎么的，情绪有点爆发了。
其实相处时间并不长，也谈不上多熟，但是却突然觉得舍不得了。

我想，是因为她的真性情，有情有义吧。
其实上班真的常常会觉得自己变得很僵硬很麻木，但是她对待生活的态度，却会让我觉得生活还是有血有肉的，很鲜活。

今天上午的时候，说到她这一年经历过的各种变故，人世间的生老病死。
也许是经历得多了，心才会变得达观，方觉真情和真诚的宝贵。

我觉得她是能够让人感受到，你给与她真诚，她会稳稳接住的人。
可以让人卸下防备，而不用担心着，觉得要保护自己。

好像上班后的一些经历，也是学着要保护自己，自己的一时心软，是会伤害到自己的。

不过要说人和人的交往，那当然还是放下防备的感觉更好。
不管它那么多的感觉更好。

想起那时候读书的时候有大小两个导师：
在大导师面前感觉是得表现得自己比较认真努力上进，比较拘束有压力；
在小导师面前就特别轻松，什么稀奇古怪的想法都可以随便说，觉得自己都变得开心了。

其实我觉得现在我心里也有类似的两个不同的声音：
一个要求自己要表现得优秀优秀、上进上进、出众出众；
一个只要率真自然就觉得开心，有些稀奇古怪的想法，不过挺自得其乐。
一个是说我要牛逼要优秀；
一个是说我开心我乐意。

我觉得有时候是不敢真性情的，
因为怕这样的自己不够好，怕别人会不喜欢这样自己；
还有一个是怕自己会受到伤害，所以要留出一点安全距离来。

所以，要走出来，就一点点的试探往外走一步，一步。
呀，原来我还是安全的。

安全的

其实自己学编程的经历，也像是在一点点的试探。

试探自己熟悉的，
试探自己未知的，
试探成功，试探错误；
试探欣喜，试探失败；
试探心血来潮飞速做作业，
试探最后一晚临时抱佛脚，
试探被激荡的心情一路飞，
试探情绪低落一步步挣扎，
试探换一种语气记录心情，
试探随时写下自己的思路...

我觉得其实都是在试探，试探我是安全的。
试探，原来即使我走到这一步，也并不会发生我以为的多么糟糕的事情，
试探，即使时间有限精力有限，我也有能力安然度过这些未知的困境。
在试探中，是一点点的建立起自己的信心，相信自己能够走过去。
即使面对低谷，面对失意，面对情绪来袭，依然可以走过去。

我慢慢意识到，对我来说，这种安全不是来自于稳定的环境，精准的计划和执行，
反而是面对混乱的时候，相信自己能穿越过去的能力的一种信心。
相信在这样的混乱的程度之下，自己依然是安全的。

就像大妈这周说到，之前没跟上，没关系，管它1w,2w,3w任务，其实现在随时都可以跟上。
当时其实在想，这样居然也可以吗？
仔细回想了一下每周的任务，好像也是这么一回事。
不过我觉得如果是我掉了一个月的进度，内心是很崩溃的，觉得追赶好累啊，心情也会很懈怠，根本不会想到，其实现在随时就能追上。我觉得我要解决N多历史遗留问题，心情很沉重。

甩掉包袱，轻松上阵。原来这样也可以！

想起以前每次自己一和别人对比，就感受到深深的差距，然后就是挫败感，自己就觉得自己不行，自己直接把自己给吓倒了。
喵，现在给我换了个思维来看一下，其实也没有自己想的那么可怕的。
深呼一口气，感受一下自己的焦虑，然后想想，如果不被这股焦虑催促会如何，真的有自己以为的那么糟糕吗？我的世界真的要崩溃了吗？真的要毁灭了吗？
想完之后，干脆小范围试验一下，如果真的做了自己担心的事情，后果真的那么糟糕吗？
放下这股焦虑，放下这股恐惧，我会有什么新的想法？有什么想真正尝试的新东西？而不是只是被这焦虑所追逐？

当我觉得我是安全的，当我相信即使在混乱中我也是安全的，我的世界会有 不同？我会看到什么新的可能性？

那不存在的门

想起年初时立下新年誓愿的时候，我觉得学编程还是一件多么需要决心和毅力的事情，觉得要跳起来才够得到。
到前三个季度的时候，我都觉得这个新年誓愿看来又要推到明年了。
谁知道，居然在第四个季度出现了转机。

原来我的新年誓愿并不需要用一年去实现，哪怕在这一年的最后100天的时候幡然醒悟，一切也都还来得及。

大妈说，入门的心理状态是错了的。
当自己入门了，happy的和python对话。
求入门，是把自己设定在门外了。

那不存在的门，之所以存在，是因为自己把自己设定在门外。

那这距离，是如何被改变的呢？
我觉得应该是从我自己真的开始琢磨着去搞定问题、去完成任务、去开始编程开始的。

换一个角度，我和音乐的距离有多远？
我觉得其实也是从自己学着弹奏乐曲、以及尝试简单创作开始。
我并不需要我以为的那么多准备工作，而是可以现在就开始。

我离我想要的生活有多远？
可不可以现在就开始？

独乐乐不如众乐乐

生命需要分享。

可以说，这是我之前基本上没怎么体会到的一个方面。
因为我一直觉得，自己喜欢的事情，一个人做也ok，反正我有时候也比较自high。
身边的人的喜好，自己也不用去干涉和影响，不要把自己的价值观加到别人上。
你喜欢你的，我喜欢我的，井水不犯河水。

我觉得其实也有点像玩音乐，风格不一样的，有时候真的不想一起玩，觉得玩不到一块。
不过有时候感觉来了，大家一起即兴真的好好玩，简直就是心情特别激荡。喵的，还是大家一起的感觉最好了。

现在互联网这么大的世界，找到一些有些相似兴趣的人，其实也不是那么难的事情嘛。
和身边的人分享，起码和哥哥分享，也是值得尝试的呀，只是之前没有想过。

这次从自己和大家的分享和反馈中真的获得很多能量，非常的超出我的预期。因为以前完全就是无视这一块嘛...

我觉得自己以后做什么事情，也会把分享和反馈这一部分给纳入进来。给自己一个好的支持系统。

Week8:休养生息

Week8:休养生息

本周，去奶奶家小住一段时间，断网。

每天晚上吃奶奶做的饭，
和奶奶看看电视，
早早睡觉，
上下班近可以走路，不用挤车。

孕吐难受，
只想快点熬过一天又一天。

虽然如此，觉得真的这样过，虽然轻松些，但是又有点空虚。
感觉还是要找点事情干，分散注意力，
不然注意力都在身体不适上了，更不舒服。

发现彩虹糖缓解孕吐有奇效，比梅子、葡萄干都好。

Week9:Go on

微信平台-小小日记

> 更新日志：
> 20151124 完成“测试存储日记”
> 20151124 创建

注册公众平台账号

微信.公众平台 https://mp.weixin.qq.com/
点击右上角进行注册
个人类型只支持注册订阅号

成为开发者

登陆“微信.公众平台”
点击左边目录的“开发者中心”
接受协议即可 在“开发者工具”里有提供开发者文档

服务器配置

阅读开发者文档的接入指南部分，
了解到是要在自己的网站上添加一段代码，
使得它能够获取微信服务器所发送的GET参数请求中携带的四个参数，
完成相关的校验流程，
返回echostr参数的内容。

具体这段代码要怎么写呢？
在Alan同学的笔记中找到了具体的实现方式。

对照接入指南中的流程，对代码进行了一下梳理与注释。

将这段代码加入到自己的网站中，
并将微信的服务器配置URL设为“http://自己的网址/wechat”，
比如我的就是http://zoejane.pythonanywhere.com/wechat

验证成功，搞定！ 启用服务器配置

代码

```
@route("/wechat")
def checkSignature():

    # 获取微信服务器所发送的GET参数请求中携带的四个参数
    signature = request.GET.get('signature', None)
    timestamp = request.GET.get('timestamp', None)
    nonce = request.GET.get('nonce', None)
    echostr = request.GET.get('echostr', None)

    token = "mytoken" # 你在微信公众平台上设置的TOKEN

    # 将token、timestamp、nonce三个参数进行字典序排序
    tmpList = [token, timestamp, nonce]
    tmpList.sort()

    # 将三个参数字符串拼接成一个字符串进行sha1加密
    import hashlib
    tmpstr = "%s%s%s" % tuple(tmpList)
    hashstr = hashlib.sha1(tmpstr).hexdigest()

    #开发者获得加密后的字符串可与signature对比，标识该请求来源于微信
    # 若确认此次GET请求来自微信服务器，请原样返回echostr参数内容，则接入生效
    if hashstr == signature:
        return echostr
    else:
        return "error"
```

Just for fun,绕过复杂的服务器配置

其实服务器的配置只是检测有没有返回echostr
所以，看似复杂的服务器配置根本也可以不用这么麻烦
只要这一段就可以了

```
@route("/wechat")
def checkSignature():
    # 获取echostr
    echostr = request.GET.get('echostr', None)
    # 返回echostr
    return echostr
```

测试通过！

所以其实写不出上一段中的那一段略微复杂的代码也没有关系，因为这个根本也不是重点...
不过我差点就卡在这里了 因为我开始的时候根本看不懂校验流程说的到底是什么 以为开发者模式很复杂

测试Echo功能

受到Alan同学的笔记的启发，测试了他写的Echo功能的代码，
也就是用户发一句话，我就可以回复同样一句话。
结合阅读“开发者文档的接收普通消息”部分，对微信的文本格式有了一个大概的了解。

而且在微信公众号上看到自己给自己的回复，感觉好好玩，哈哈，有种自己给自己做了一个玩具的感觉。

对于字典这一部分，还有些不了解，需要补课。
考虑这几天把Learn python the hard way的相关部分补完。

不过既然都可以echo了，那我也就可以测试存储日记啦。

测试存储日记

把我之前写的写日记的相关代码添加进来。

将Alan同学代码的这一部分，

```
# 更新时间
import time
mydict['CreateTime'] = int(time.time())

# 现在不对内容做任何操作，只是原样返回
```

改写成现在我想用的新功能

```
# 添加日记
today=datetime.now()
newDiary=mydict['Content']
user_name=mydict['FromUserName']

with open('diary.txt', 'r+') as f:
    content = f.read()
    f.seek(0, 0)
    newDiaryLine=today.strftime("%Y/%m/%d/ %T")+ '  ['+user_name+'] '+newDiary
    f.write(newDiaryLine.rstrip('\r\n') + '\n' + content)

# 更新时间
import time
mydict['CreateTime'] = int(time.time())
# 更新回复内容
mydict['Content'] = mydict['Content']+'已保存'
```

就实现了两个小小功能：

1. 将用户发送的内容自动添加到diary.txt中
2. 回复用户 ‘XXX已保存’

其实有了mydict['Content']这部分内容，就会发现和第一周命令行的开发差不多了

增加帮助和阅读功能，增加中文支持

用户输入'help',可以看到帮助
输入'read'，可以阅读历史日记

之前有碰到输入中文就会出现‘暂时无法提供服务的现象’,
后来来了解到,在接受消息和发送消息时，消息的encode要改成UTF-8

```
# 添加帮助
if mydict['Content'] =='help':
    mydict['Content'] = '''
    输入“help”或者“帮助”可以看到帮助
    输入“read”或者“阅读”可以阅读历史日记
    '''

# 添加阅读
elif mydict['Content'] =='read':
    diaryFile = open('diary-wechat.txt')
    diaryContent = diaryFile.read()
    diaryFile.close()
    mydict['Content'] = diaryContent

else:
    # 添加日记
    today=datetime.now()
    newDiary=mydict['Content'].encode('UTF-8')

    with open('diary-wechat.txt', 'r+') as f:
        content = f.read()
        f.seek(0, 0)
        newDiaryLine=today.strftime("%Y/%m/%d/ %T")+ ' '+newDiary
        f.write(newDiaryLine.rstrip('\r\n') + '\n' + content)

    # 更新时间
    import time
    mydict['CreateTime'] = int(time.time())
    # 更新回复内容
    mydict['Content'] = mydict['Content'].encode('UTF-8')+'已保存'
```

添加多用户支持

根据FromUserName来识别，给每个用户建立专属的FromUserName.txt文件

```
user_name=mydict['FromUserName']
user_diary=str(user_name)+'.txt'

# 添加阅读
elif mydict['Content'].lower() =='read':
    diaryFile = open(user_diary)
    diaryContent = diaryFile.read()
    diaryFile.close()
    mydict['Content'] = diaryContent

else:
    # 添加日记
    today=datetime.now()
    newDiary=mydict['Content'].encode('UTF-8')

    diaryFile = open(user_diary,'a')
    diaryFile.close()

    with open(user_diary, 'r+') as f:
        content = f.read()
        f.seek(0, 0)
        newDiaryLine=today.strftime("%Y/%m/%d/ %T")+ ' '+newDiary
        f.write(newDiaryLine.rstrip('\r\n') + '\n' + content)'
```

修正时区

```
from datetime import datetime
import pytz

today = datetime.now(pytz.timezone('Asia/Shanghai'))
```

感想

- 谢谢大家的教程和经验，让我跳过了好多纠结的地方
- 本来以为服务器验证很难，后来发现其实根本无关紧要，直接return echostr绕过去都可以
- 微信接口打通后，发现后面写的内容和第一周的命令行版本的编程差不多了
- 第一次在微信上自己和自己的公众号对话的时候，感觉很奇妙，哈哈，好玩
- 好像休息了10多天了，继续爬过来交作业

另外：我的公众号是希望每个人都有个独立的日记本，可以不把很多人的日记混在一起。不知道这个功能是不是正常，还请大家测试一下，谢谢啦。

为自己建立更大的支持系统

想一想，这次遇到的难题是如何解决的

微信的服务器验证和消息传输，最开始看起来，就像一个坑，说的都是中文，但我完全不知道它在说什么。

感谢Alan同学的笔记，感谢大家提供的无私教程，感谢摸着石头过河的大家提供的经验。
当碰到困难，我一时又解决不了的时候，我知道，去别人教程那里串门一下，很多问题就解决了。

但如果是自己一个人学呢，可能几次搜索无果，几次询问无果后，就不了了之了。
这也是自学容易碰到的问题。

其实很多东西我根本不懂，我只是模仿别人的经验，尝试，误打误撞的成了之后，我才逐渐明白之前做的到底是什么意思。
可以说是先做成了之后，再回过头来慢慢理解的。
然后会发现，之前觉得难的事情，也不是那么难了。

问题是，自学的时候，可能并没有这么好的支持系统，让你正好可以找到解决方案。

包括自己上班也有这样的感觉，不知道的问题问同事，问主管，当然现在有些内容是可以有知识系统去搜索，有员工热线去协助（也是基于知识系统的）。
但是如果他们都不知道呢，这个问题就一直放在那里悬而未决了。
可能解决方案并不难，但是因为我请教的这个范围内的人都没人知道，就好像突然断了线索一样了。

也是因为这样，我发现很多问题并不能只是问同事、问前辈，业务繁杂，变化太快，记忆有限。
必须要给自己一个更大的支持系统。 不能仅限于自己能够直接接触到和观察到的人，因为直接接触到的人太有限，这个世界真的太狭窄。

重新梳理工作的支持系统

1. 业务知识支持

系统内

- 基本的手册和教材 - 搭建本部门基本框架
- 系统里的知识库 - 日常问题的搜索
- 员工热线 - 解决较为紧急的疑难问题
- 网络学习系统 - 能了解各个部门的一些基础知识

- 官方网站，包括年报等 - 了解总部最新发展方向

系统外

- 在别的我感兴趣的行业，领域里，有没有用到我这个工作领域的知识的？
- 他们是如何运用这种知识的？
- 我该如何利用我自己的这个平台，积累这种领域知识，去做些更有趣的事情？
- 本领域有趣的大牛是什么样的，能从他们身上学到什么？

2.软技能支持 - 这也是更加跨领域的能力

- 学习能力
- 书面表达能力 如写作
- 口头表达能力 如日常交流，演讲
- 创造力
- 数据处理能力
- 信息和知识管理能力
- 逻辑能力
- 销售能力
- 战略部署能力
- 研究分析能力
- 团队协作能力
- ……

再扯可以扯很远了。

我觉得其实这些软技能的学习的来源是可以更广泛的，因为这本来就是更跨领域的能力，也就是说在各个领域中，都可以汲取到好的方式和经验。
只局限于自己的工作领域，会太狭隘。

可以把自己的工作领域，当成这些技能的一个练习和展示的平台。
但是视野一定要更广阔一些，跳出现有的工作领域之外。

如果我要开启新的学习

通过这次的学习过程，我觉得我要多问几个自己之前并没有特别意识到的问题。

1. 有没有一个好的反馈系统？
   除了自己自己记录自己成长和摸索，有没有更多的一些外部的互动？
   让你的每一次进步能够让你有所鼓舞，你的错误也能够及早纠正？
   让你能看到和你不一样的视角？
   比如可以一起交流的伙伴，可以督促和鼓励你的人，能够指出你一些误区的人，比你更有经验的人？

2. 有没有一个比较清晰的目标和截止时间？
   防止不知道要干嘛,
   防止无限期的拖延。

3. 有没有一个好的支持和帮助系统？
   如果碰到了问题，难题，能够给与我比较好的支持？
   而不是让我卡住了，停住了。

比如我想学音乐。
那么我想达到一个什么样的目标，达到什么情况就算是实现了我的这个目标？
我想在什么时间里完成这件事情？
我怎么知道自己的失误？
怎么鼓励自己的进步？
能不能找到和我一起学习的小伙伴？
能不能找到好的学习社区或者导师？
如果我碰到难题，我有一些什么样的资源可以帮助我跨越这个障碍？

迭代开发

~ 此目录收集, 持续开发时各种体验

进展

- 150924 大妈创建

Week10:What am I wanna do?

What am I wanna do

时间不知不觉，走到了即将该组队的时刻了。

我的脑袋里有股冲动的情绪，有时候却又觉得空空如也。
我不知道我想做什么，我也不知道要如何开始组一个队。

我好像可以预设一万种困难。
如果我想不出想做的事情怎么办？
如果我写不出来怎么办？
如果我想组队的人和别人组队更方便怎么办？
我不在三大区不能face to face怎么办？
在团队里我能做什么呢？ ……

但是我又想尝试啊，都已经来到这里了，难道还要一个人编程吗？
我想大家一起玩。

就像那时候玩音乐的时候觉得，大家一起玩才最最开心了。
就像那时候想的，有一个小小团队，大家一起来做一件我们都喜欢的事情，该是多么快乐。

我感觉到时光的流逝，
我不知道我还有多少时间，
我更不知道我还有多少勇气。

我只想走到生命的尽头的时候，能觉得不枉此生。
能觉得那些我想做的事情，我都去做过，去尝试过，去靠近过。
能觉得我一直没有放弃自己喜欢的生活，一直为自己的眼里留下一抹闪亮的光。

大不了就是被拒绝，大不了就是失败。
可是如果我不开始去做，
我要如何去靠近呢？

等待，是等待不来梦想的召唤。
我想我要学会去创造生命中的一些惊喜。

这次失败了，我也能承受，也不是什么大不了的失败。
而不去开始的失败，不去为自己喜欢的生活去努力尝试的生活，这样的结局才是我真正不能承受的吧。

我不想觉得就是不甘心，就是不服气，可是我已经无能为力了，我已经被磨灭的无能为力了，使不出力气了。
我真的很害怕会变成这样，我痛恨现在的生活，却已经没有力气去改变了。

不是真的没有力气去改变，而是已经被驯养太久，已经不会使出自己的力气了。
明明可以做到的，却觉得自己做不到，还没有开始就直接放弃了。

我想到这个世界曾为我推开的那些门。
那些曾让我觉得特别快乐的时刻。
满心欣喜，觉得梦想很近的时刻。
觉得即使前路艰难险阻，我也要走下去的时刻。

想起那时候，也有过那样乐观和有勇气的时刻。

本来已经觉得要忘记了，但是不知怎么最近又老是想起来。
也许是冬天太冷，靠近死亡的气息，就让人越发的想起那些生机盎然，想起生命里那些闪亮的时刻。

我不想看到自己以前写的东西就痛哭，我也不想听到自己以前做过的音乐就痛哭，不想想起以前的日记就痛哭，不想觉得现在是对以前的背叛。

我想做一件事情，想一直做一件事情，
就是提醒自己向自己真正喜欢的人生方向去靠近，
去做一些自己内心真正喜欢的，真正觉得开心很满足的事情，

我想这次我想用python做的，也是类似的东西。
能够提醒我生命中真正渴望的事情是什么，
能够给我勇气去继续，
能够在我失落和陷入低谷的时候给予我支持，让我度过难关， 能够在我思维卡壳的时候，给我启发，给我能量，让我往下走下去， 能够在纷纷扰扰中，不忘记生命重要的事情。

也许就像是个梦想的秘密花园吧，或许也是梦想的墓志铭。
写下我的梦想，写下我的情绪，写下我要如何去靠近，写下我进步的每一天。

Part1 黑 死亡 生命宝贵 我剩下的时间不多 很少 多少个周末 多少个新年 都数的过来

Part2 粉 希望 我的梦想清单
Part3 绿 生命 前进的每一天 我的日记
Part4 天蓝 自由 鼓励自己的一些东西
Thinking:

- 可以考虑合并 都是日记 tag：梦想（不过这个一篇就够了） tag：行动日记 tag：鼓励 能量）
- Milestone 到点还是要给自己一个奖励吧 不能无限制努力吧 每前进一步就要给与奖励 也要给自己一点成功的反馈
  考虑用每周的形式 或者自己设置不同的小目标 决定添加一个part 黄 成就 Milestone
- 要有搜索功能
- 游戏化，游戏为什么能坚持

Part5 白 连接 人 支持 反馈 互动 可以邀请朋友，自己信任的人一起参与这个过程 评论 互动

梦想的秘密花园

缘起

人生有很多梦想，但也许走到后来也未曾实现，就这样渐渐湮没在每一天里。
给自己一个小小的属于梦想的秘密花园，提醒自己，人生的时光有限，为自己留一片属于自己的秘密花园。
记录前进的每一天，记录快乐与低谷，记录分享，记录感动。

设计

Part1 黑 死亡 生命宝贵

显示我所剩下的时间。
总以为生命漫长，在这里，面对生命的倒计时，才会知道，原来，有多少个周末，多少个新年， 都数的过来。
原来我并没有那么多的以后，要从现在开始。

Part2 粉 希望 我的梦想清单

写下我想完成的梦想，心愿

Part3 绿 生命 前进的每一天 我的日记

记录我生活的每一天，为梦想前进的每一天。
无论高兴或是低谷，这都是我生命的一部分。

Part4 黄 里程碑 创作集 值得纪念的一些时刻

记录自己的作品和创作，记录自己的改变，记录自己做到了以前没有做到的事情，让自己觉得兴奋和快乐的时刻，让自己觉得成长的时刻。

Part5 蓝 自由 鼓励自己的一些东西

记录给自己有启发的一些文章，让自己振奋的音乐，一切喜欢的图片，一段好的演讲等。

Part6 白 连接 人 支持 反馈 互动

可以邀请朋友，自己信任的人一起参与这个过程，评论，互动。 可以探索他人的秘密花园，交流，鼓励，互助。

形式

- ios App - 希望自己的手机能这个小程序
- Web版本 - 如果没有手机，也还是可以用别的方式打开

问题

1.软件功能太散，现有开发能力完全hold不住。

感觉列出来的功能太多了。
总觉得应该先集中在一个很小的点去突破会更好。
但是一下子不知道往哪里集中。

2.ios开发貌似用python进行不了，或者很找虐。

折衷的方式是想办法开发web或者微信版本。
或者干脆学Swift，因为确实想看看做个app是个什么感觉。

3.感觉做出来了自己也不会用，因为有很多优秀的app可以更好地实现某些功能。

比如自己虽然一直在练习写小小笔记系统，但是自己实际记录笔记还是在evernote上。
觉得开发出来的软件太弱，开发出来了自己也不想用。
喵，实现了基本功能又如何，还是不想用不想用。

实际

1. 觉得更靠谱的方式，是把这几个part当成自己的一个思考系统。
   先别管用不用python编程了，如果能形成一个能帮助自己一点点靠近梦想的思考系统也很好呀。
2. 顺便去发掘一些好用的相关app和学习一些相关文章。看看有什么新的启发，能让我找到一个更小的突破口。
3. 还有一个想法是，观察自己和采访他人在这次学习过程中的体验，写成文章？做个PPT？之类之类的。写一份这一次的学习报告，看看让大家坚持下来的原因有什么，开心的地方有什么。

4. 看看有什么大家的项目自己可以加入的，体验一下协作的氛围。
5. Feedback很关键，如何为自己打造一个Feedback和支持系统？我觉得这是很多学习能否继续的关键因素。

让我开心的三件事

收到培炎教练的鼓励

> 培炎：
> 虽然注重过程的心态可嘉，但我不能完全同意，
> 你对自己的项目问题认识得非常到位，最主要的是功能太杂，只要坚定认准一个点，是完全可以做出来的。
> 建议选择有特色的点入手，有成熟替代品的尽量不要选择。
> 我个人非常喜欢黑色创意，可实现性也非常高，
> 如果嫌简单，其实深入进去也有可挖掘处，
> 比如说：怎么设定自己生命的长度？可否记录自己的生活习惯，使这个长度随习惯改变可变，从而督促戒掉坏习惯？
> 只有周末和新年重要吗？亲人的生日、结婚纪念日？人生理想的阶段目标？是否可以加入，从而成为一种超长程的计划管理软件？ ……等等吧，仔细做起来，我想也是非常精彩的，期待能看到你的路演展示

我：
喵，看得我太感动了。 确实，我觉得之前想开发又没动力开发的原因，是因为成熟的替代品很多，导致我不想去用。 所以切入点是要从这里去入手，这是我之前所没有想到的。
相对来说，黑色系列的作品是比较少的，但是确实也是让我印象深刻的体验到，原来时间这么少。而且这系列的软件主要集中在倒计时上，而其实时间的长度也不代表时间的质量。这里确实有很多可以思考的地方。
最重要的是，我觉得在思考这个软件设计的同时，也是在思考自己如何度过自己的人生，这种感觉很神奇。
真的非常谢谢你的指导，包括你之前和别人的对话，都非常有启发。希望以后有更多的机会向你学习:)

加入情绪日记的团队

真的加入一个小团队，感觉很兴奋。
晚上讨论得很开心，有团队和一个人单打独斗的感觉很不一样。
组长WhaleChen很用心很负责很细致，觉得能从他身上学到很多。
很喜欢有个团队的感觉。

免费加入**Udacity Guru**

Udacity看视频本来看得我有些打瞌睡了的，囫囵吞枣的交了作业后，意外收到了很细致的review，突然感觉很不同。他细致的解释，让我意识到了自己之前思考的问题所在，也让我想要好好的去重新学好这一部分了。
这也让我认识到了反馈的强大作用。
正好这两天在想Feedback的作用，以及如何持续收到Feedback的事情，没想到意外收到Udacity的来信，直接就给我提供了这样的机会，这感觉真的太神奇了，特别开心，真有种天降好运的感觉。
慢慢觉得这个探索过程越来越有趣了。

> What is Udacity Guru?
>
> The program pairs students like yourself with an experienced iOS developer, a Guru, who will provide feedback and guidance throughout your learning experience. This means you get one dedicated person who is invested in your success, who can answer your questions, troubleshoot with you when things go wrong, and celebrate the milestones with you.

产品路演

~ 此目录收集, 最后一阶段的体验

进展

- 150924 大妈创建

OMOOC.py 实践代码目录

~ 收集每周任务代码

草案/原型

~ 此目录收集, 还在探索/思考 中的文文案

成熟后应该单立文档目录 长期增补

进展

- 150316 大妈创建

教程应该怎么写?

从邮件中恢复

```
发件人：    Zoom.Quiet <zoom.quiet@gmail.com>
收件人：    "omooc@googlegroups.com" <omooc@googlegroups.com>
日期： 2015年3月17日 下午2:44
主题： [INFO]教程怎么写?
```

- 教程怎么去写?
- 面向 6 个月前的自己来写吧!
    - 软件工程中有一个著名的沟通技巧: 小黄鸭方法
    - 也同样适用于写教程 ;-)
    - 因为, 我们对自个儿的知识结构最了解
    - 最知道怎么说服这样的一个自个儿来使用具体的技术
    - 或是说,最能直觉的整理出这样的自己,如何学习最效!
- 但是,大妈呢, 作为第一读者关注的则是:
    - 是否原创?
    - 如果是抄的话,抄的姿势对不对?
    - 应用这篇教程,是否能说服同类小伙伴来使用?
    - 教程是否完备/可用
    - 简单就的说就是 DUKU 6字方针:
        - 有料
        - 有趣
        - 有种

是也乎

- 150412 大妈创建

有关

~ 汇集这部私人教程未尽体验

TODO

参考

- 图灵社区：阅读：使用GitBook平台发布电子书
- 使用Gitbook来编写你的Api文档 « 悟道集

(￣▽￣)

- 150924 ZQ 清理为学员模板
- 150317 迁移用 开智官方 gitbook 身份
- 150316 ZQ 创建